aws

Guida per gli sviluppatori

AWS Key Management Service

AWS Key Management Service: Guida per gli sviluppatori

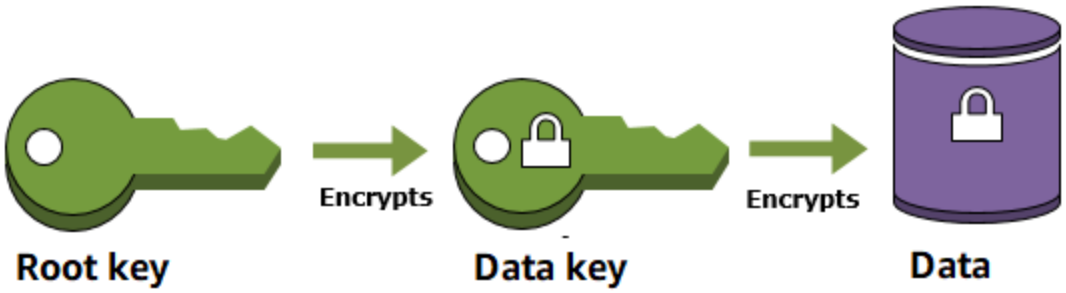
Table of Contents

AWS Key Management Service

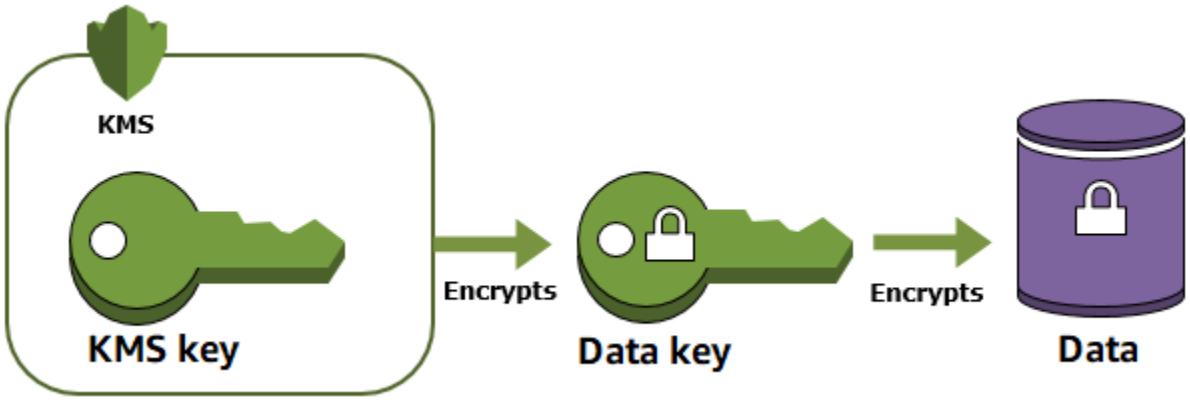
AWS Key Management Service (AWS KMS) è un servizio AWS gestito che semplifica la creazione e il controllo delle chiavi di crittografia utilizzate per crittografare i dati. I AWS KMS keys file creati AWS KMS sono protetti da FIPS140-2 moduli di sicurezza hardware convalidati (). HSM Non escono mai senza crittografia AWS KMS . Per utilizzare o gestire le tue KMS chiavi, 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. KMSle chiavi vengono create, gestite, utilizzate ed eliminate interamente all'interno AWS KMS. Non lasciano mai il servizio senza crittografia. Per utilizzare o gestire le tue KMS chiavi, AWS KMS chiami.



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

AWS KMS nel 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 utilizzare AWS KMS nei seguenti modi:

AWS Management Console

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 AWS Management Console e selezionando AWS KMS dalla AWS Management Console 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 AWS KMS autorizzazioni, gli utenti devono disporre anche delle autorizzazioni per elencare IAM utenti e ruoli. IAM Se crei una policy IAM più restrittiva delle autorizzazioni minime richieste, la console AWS KMS non funzionerà nel modo 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 KMS le chiavi nella console AWS KMS.

Per consentire agli utenti di utilizzare la AWS KMS console per creare e gestire KMS le chiavi, allega la policy AWSKeyManagementServicePowerUsergestita all'utente, come descritto in. AWS politiche gestite per AWS Key Management Service

Non è necessario concedere autorizzazioni minime per la console agli utenti che utilizzano la console AWS KMS API tramite AWS SDKsAWS Command Line Interface, o AWS Tools for PowerShell. Tuttavia, è necessario concedere a questi utenti l'autorizzazione per utilizzare ilAPI. 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 attività AWS (incluse AWS KMS).

Per ulteriori informazioni sull'utilizzo AWS KMS tramite AWS CLI, consulta la Guida ai AWS CLI comandi

AWS KMS REST API

L'architettura di AWS KMS è progettata per essere indipendente dal linguaggio di programmazione e utilizza AWS interfacce supportate per archiviare e recuperare oggetti. È possibile accedere a S3 e a livello di codice utilizzando il AWS AWS KMS REST API. La REST API è un'HTTPinterfaccia per AWS KMS. Con il plugin REST API, si utilizzano HTTP richieste standard per creare, recuperare ed eliminare bucket e oggetti.

Per ulteriori informazioni sull'utilizzo di AWS KMS REST API, vedi il AWS Key Management Service APIriferimento

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 SDKsForniscono un modo conveniente per creare l'accesso programmatico a e. AWS KMS AWS AWS KMS è un REST servizio. È possibile inviare richieste all' AWS KMS utilizzo delle AWS SDK librerie, che racchiudono le librerie sottostanti AWS KMS REST API e semplifica le attività di programmazione. Per informazioni sul AWS SDKs, incluso come scaricarli e installarli, vedi Tools to Build on AWS.

Esempi di codice per AWS KMS l'utilizzo AWS SDKsFornisce 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 popolari. Ciascuno di essi SDK fornisceAPI, esempi di codice e documentazione che semplificano agli sviluppatori la creazione di applicazioni nel linguaggio preferito.

Documentazione SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice

Documentazione SDK	Esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for 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 Provide feedback (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 aKMS, ma si integra o può essere utilizzato come soluzione autonoma SDK senza riferimenti alle chiavi. AWS KMS 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 IAM politiche

AWS KMS eventuale coerenza

AWS KMS APISegue 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 eseguiti successivamente.

Quando si eseguono AWS KMS API chiamate, 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 API le chiamate relative alle sovvenzioni, puoi utilizzare un token di concessione per evitare potenziali ritardi e utilizzare immediatamente le autorizzazioni incluse in una concessione. Per ulteriori informazioni, consulta Consistenza finale (per concessioni).

Utilizzo del post-quantum TLS 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 TLS opzione quando ci si connette agli AWS KMS API endpoint. Offriamo questa funzionalità prima che gli algoritmi post-quantistici vengano standardizzati, quindi puoi iniziare a testare l'effetto di questi protocolli di scambio di chiavi sulle chiamate. AWS KMS Queste funzionalità opzionali di scambio di chiavi ibride post-quantistiche sono sicure almeno quanto la TLS crittografia che utilizziamo oggi e probabilmente offriranno ulteriori vantaggi di 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 suite di crittografia classiche che AWS KMS supportano TLS le sessioni rendono gli attacchi di forza bruta ai meccanismi di scambio di chiavi irrealizzabili con la tecnologia attuale. Tuttavia, se l'informatica quantistica su larga scala

diventerà pratica in futuro, le classiche suite di crittografia utilizzate nei meccanismi di scambio di TLS chiavi saranno suscettibili a questi attacchi. Se state sviluppando applicazioni che si basano sulla riservatezza a lungo termine dei dati trasmessi tramite una TLS connessione, dovreste prendere in considerazione un piano per migrare 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 TLS connessione sia almeno altrettanto forte 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 ibrida sono diversi da quelli dei classici meccanismi di scambio di chiavi, consigliamo di testarli durante le chiamate in condizioni diverse. AWS KMS API

Feedback

Come sempre, la tua opinione e la partecipazione nei nostri repository open source è molto importante. In particolare, ci piacerebbe sapere come la vostra infrastruttura interagisce con questa nuova variante di 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 cifratura ibride in formato open source nel s2n-tlsrepository su. GitHub Per fornire feedback sull'usabilità delle suite di crittografia o condividere nuove condizioni o risultati dei test, crea un problema nel s2n-tls archivio.
- Stiamo scrivendo esempi di codice per l'utilizzo del post-quantum TLS ibrido con in AWS KMS aws-kms-pq-tls-example GitHubdeposito. Per porre domande o condividere idee sulla configurazione HTTP del client o del AWS KMS client per l'utilizzo delle suite di crittografia ibrida, create un problema nel aws-kms-pq-tls-example archivio.

Supportato Regioni AWS

Post-Quantum TLS for AWS KMS è disponibile in tutti i paesi Regioni AWS AWS KMS supportati ad eccezione di Cina (Pechino) e Cina (Ningxia).

> **Note**
>
> AWS KMS non supporta il TLS post-quantum ibrido per gli endpoint in. FIPS AWS GovCloud (US)

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 informazioni sugli FIPS endpoint, consulta FIPS endpoints in. 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 HTTP client che utilizza queste suite di crittografia. Quindi, ogni volta che ti connetti a un AWS KMS endpoint con il tuo HTTP client, vengono utilizzate le suite di crittografia ibride.

Questo client utilizza HTTP s2n-tls, che è un'implementazione open source del TLS protocollo. La crittografia ibrida si adatta a s2n-tls gli usi sono implementati solo per lo scambio di chiavi, non per la crittografia diretta dei dati. 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 che s2n-tls utilizza un ibrido che combina Elliptic Curve Diffie-Hellman (ECDH), un classico algoritmo di scambio di chiavi utilizzato oggi in, con TLS Kyber, un algoritmo di crittografia e creazione di 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. 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 HTTP client in modo che preferisca il post-quantumTLS, che colloca con ECDH Kyber primo 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.

Se la ricerca in corso rivela che Kyber all'algoritmo manca la forza post-quantistica prevista, la chiave ibrida è ancora potente almeno quanto la ECDH chiave singola attualmente in uso. Fino al completamento della ricerca sugli algoritmi post-quantistici, si consiglia di utilizzare algoritmi ibridi, piuttosto che utilizzare esclusivamente algoritmi post-quantistici.

Utilizzo del post-quantum ibrido con TLS AWS KMS

È possibile utilizzare la tecnologia post-quantistica ibrida TLS per le chiamate a. AWS KMS Quando configuri l'ambiente di test del HTTP client, tieni presente le seguenti informazioni:

Crittografia in transito

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

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

Sistemi supportati

Utilizzo delle suite di cifratura ibride in s2n-tls è attualmente supportato solo su sistemi Linux. Inoltre, queste suite di crittografia sono supportate solo se SDKs supportano AWS Common Runtime, ad esempio. AWS SDK for Java 2.x Per vedere un esempio, consulta Configura la postquantistica ibrida TLS.

AWS KMS Endpoints

Quando si utilizzano le suite di crittografia ibride, utilizzare l'endpoint standard. AWS KMS Le suite di crittografia ibrida in s2n-tls non sono compatibili con gli endpoint convalidati FIPS 140-2 per. AWS KMS

Quando si configura un HTTP client per preferire le connessioni post-quantistiche con TLS s2n-tls, i cifrari post-quantistici sono i primi nell'elenco delle preferenze di cifratura. Tuttavia, l'elenco delle preferenze include le crittografie classiche non ibride in posizioni inferiori nell'ordine di preferenza per la compatibilità. Quando configuri un HTTP client per preferire la tecnologia TLS post-quantistica con un endpoint convalidato 140-2, AWS KMS FIPS s2n-tls negozia un cifrario di scambio di chiavi classico e non ibrido.

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 informazioni sugli FIPS endpoint, consulta FIPS endpoints in. Riferimenti generali di Amazon Web Services

Prestazioni previste

I nostri primi test di benchmark mostrano che le suite di crittografia ibrida sono disponibili in s2n-tls sono più lente delle suite di crittografia classiche. TLS L'effetto varia in base al profilo di rete, alla CPU velocità, al numero di core e alla frequenza delle chiamate. Per i risultati dei test delle prestazioni, vedi Come ottimizzare TLS la crittografia post-quantistica ibrida con Kyber.

Configura la postquantistica ibrida TLS

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

Per vedere alcuni esempi operativi completi di configurazione e utilizzo della tecnologia post-quantistica ibrida TLS con AWS KMS, consulta la aws-kms-pq-tls-examplearchivio.

> **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 `2.20.0` del client AWS Common Runtime alle dipendenze Maven.

   ```
   <dependency>
       <groupId>software.amazon.awssdk</groupId>
       <artifactId>aws-crt-client</artifactId>
       <version>2.20.0</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 cifratura post-quantistica ibrida sul tuo client, come mostrato nell'esempio seguente. HTTP

Questo codice utilizza il parametro `postQuantumTlsEnabled()` method per configurare un HTTPclient di runtime AWS comune che preferisce la suite di cifratura post-quantistica ibrida consigliata, con Kyber. ECDH Quindi utilizza il HTTP client configurato per creare un'istanza del client asincrono,. AWS KMS `KmsAsyncClient` Una volta completato questo codice, tutte le AWS KMS APIrichieste sull'`KmsAsyncClient`istanza utilizzano la tecnologia post-quantistica ibrida. TLS

```
// 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 la tecnologia post-quantistica ibrida. TLS

   Quando richiami AWS KMS API le operazioni sul AWS KMS client configurato, le chiamate vengono trasmesse all' AWS KMS endpoint utilizzando la tecnologia post-quantistica ibrida. TLS Per testare la configurazione, chiama un AWS KMS API, ad esempio. `ListKeys`

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

Testa la tua configurazione ibrida post-quantistica TLS

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

- Eseguire test di carico e benchmark. Le suite di crittografia ibrida funzionano in modo diverso rispetto agli algoritmi di scambio di chiavi tradizionali. Potrebbe essere necessario modificare i timeout della connessione per consentire tempi di handshake più lunghi. Se esegui all'interno di una AWS Lambda funzione, estendi l'impostazione del timeout di esecuzione.
- Provare a connettersi da posizioni diverse. A seconda del percorso di rete seguito dalla richiesta, potresti scoprire che host intermedi, proxy o firewall con deep packet inspection () bloccano la richiesta. DPI Ciò potrebbe derivare dall'utilizzo delle nuove suite di crittografia nella ClientHelloparte dell'TLShandshake 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 le nuove suite di crittografia. TLS

Scopri di più sul post-quantum in TLS AWS KMS

Per ulteriori informazioni sull'utilizzo della tecnologia post-quantistica ibrida AWS KMS, TLS consultate 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, vedere Crittografia post-quantistica.
- Per informazioni su s2n-tls, vedere Introduzione s2n-tls, una nuova TLS implementazione e utilizzo dell'open source s2n-tls.
- Per informazioni sul AWS Common Runtime HTTP Client, vedere Configurazione del HTTP client AWS CRT basato nella Guida per gli AWS SDK for Java 2.x sviluppatori.
- Per informazioni sul progetto di crittografia post-quantistica presso il National Institute for Standards and Technology (NIST), vedere Post-Quantum Cryptography.
- Per informazioni sulla standardizzazione della crittografia NIST post-quantistica, vedete Post-Quantum Cryptography Standardization.

Connect a AWS KMS tramite un VPC endpoint

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

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

L'VPCendpoint dell'interfaccia si connette VPC direttamente all'utente AWS KMS senza un gateway Internet, un NAT dispositivo, VPN una connessione o una connessione. AWS Direct Connect Le istanze presenti VPC non necessitano di indirizzi IP pubblici con cui comunicare. AWS KMS

Regioni

AWS KMS supporta VPC gli endpoint e le policy relative agli VPC endpoint Regioni AWS in tutte le aree supportate AWS KMS.

Considerazioni per gli endpoint AWS KMS VPC

Prima di configurare un VPC endpoint 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 VPC endpoint include quanto segue.

- Puoi usare il tuo VPC endpoint per chiamare tutte le AWS KMS APIoperazioni dal tuo. VPC
- È possibile creare un VPC endpoint di interfaccia che si connette a un endpoint o a un endpoint AWS KMS regionale.AWS KMS FIPS
- È possibile utilizzare AWS CloudTrail i log per verificare l'utilizzo delle KMS chiavi tramite l'endpoint. VPC Per informazioni dettagliate, consultare Registrazione delle AWS KMS richieste che utilizzano un endpoint VPC.

Argomenti

Crea un endpoint per VPC AWS KMS

Puoi creare un VPC endpoint per AWS KMS utilizzando la VPC console Amazon o Amazon VPCAPI. Segui le procedure per creare un endpoint di interfaccia utilizzando uno dei seguenti valori.

- Per creare un VPC endpoint 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 VPC endpoint che si connette a un AWS KMS FIPSendpoint, 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'VPCendpoint, puoi abilitare un DNSnome privato per l'endpoint. VPC Se selezioni l'opzione Enable DNS Name, il AWS KMS DNS nome host standard viene risolto nel tuo endpoint. VPC Ad esempio, si `https://kms.us-west-2.amazonaws.com` risolverebbe in un VPC endpoint connesso al nome del servizio. `com.amazonaws.us-west-2.kms`

Questa opzione semplifica l'utilizzo dell'VPCendpoint. Per impostazione predefinita, AWS CLI utilizza il AWS KMS DNS nome host standard, quindi non è necessario specificare l'VPCendpoint URL 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 VPC endpoint

È possibile connettersi AWS KMS tramite l'VPCendpoint utilizzando un AWS SDK, il AWS CLI, o. AWS Tools for PowerShell Per specificare l'VPCendpoint, usa il suo DNS nome.

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

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcdf5678c90a-09p7654s-us-
east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Autorizzazioni richieste

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

- Una politica, una IAMpolitica o una concessione chiave deve fornire l'autorizzazione principale a richiamare l'operazione sulla risorsa (KMSchiave o alias).

- Una policy relativa all'VPCendpoint deve fornire l'autorizzazione principale per utilizzare l'endpoint per effettuare la richiesta.

Ad esempio, una policy chiave potrebbe fornire l'autorizzazione principale per chiamare Decrypt su una chiave particolare. KMS Tuttavia, la policy dell'VPCendpoint potrebbe non consentire a quel principale di `Decrypt` richiamare quella KMS chiave utilizzando l'endpoint.

Oppure una policy VPC sugli endpoint potrebbe consentire a un principale di utilizzare l'endpoint per DisableKeyrichiamare determinate chiavi. KMS Ma se il preside non dispone delle autorizzazioni previste da una policy, IAM policy o concessione chiave, la richiesta ha esito negativo.

Puoi creare una policy per gli VPC endpoint quando crei l'endpoint e puoi modificare la policy degli VPC endpoint in qualsiasi momento. Utilizza la console VPC di gestione o le CreateVpcEndpointoperazioni. ModifyVpcEndpoint È inoltre possibile creare e modificare una policy per gli VPC endpoint utilizzando un AWS CloudFormation modello. Per informazioni sull'utilizzo della console di VPC gestione, consulta Creare un endpoint di interfaccia e Modificare un endpoint di interfaccia nella Guida.AWS PrivateLink

Nomi host privati

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

Per utilizzare i nomi di host privati, `enableDnsSupport` gli attributi `enableDnsHostnames` e del tuo VPC devono essere impostati su. `true` Per impostare questi attributi, usa l'ModifyVpcAttributeoperazione. Per i dettagli, consulta Visualizza e aggiorna DNS gli attributi per i tuoi VPC nella Amazon VPC User Guide.

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

Puoi controllare l'accesso alle AWS KMS risorse e alle operazioni quando la richiesta proviene VPC o utilizza un VPC endpoint. A tale scopo, utilizzate una delle seguenti chiavi di condizione globali in una policy o IAMpolicy chiave.

- Utilizza la chiave `aws:sourceVpce` condizionale per concedere o limitare l'accesso in base all'VPCendpoint.
- Usa la chiave `aws:sourceVpc` condizionale per concedere o limitare l'accesso in base a VPC quello che ospita l'endpoint privato.

> **Note**
>
> Fai attenzione quando crei policy e IAM policy chiave basate sul tuo VPC endpoint. Se una dichiarazione di policy richiede che le richieste provengano da un particolare VPC o da un VPC endpoint, 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 VPC degli endpoint nelle politiche con autorizzazioni AWS KMS.
> Inoltre, la chiave di `aws:sourceIP` condizione non è efficace quando la richiesta proviene da un VPCendpoint Amazon. Per limitare le richieste a un VPC endpoint, usa le chiavi di `aws:sourceVpc` condizione `aws:sourceVpce` o. Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per VPC endpoint e servizi VPC endpoint nella Guida.AWS PrivateLink

Puoi utilizzare queste chiavi di condizione globali per controllare l'accesso a AWS KMS keys (KMSchiavi), alias e operazioni del genere CreateKeyche non dipendono da alcuna risorsa particolare.

Ad esempio, la seguente politica di chiave di esempio consente a un utente di eseguire alcune operazioni crittografiche con una KMS chiave solo quando la richiesta utilizza l'endpoint specificatoVPC. Quando un utente effettua una richiesta a AWS KMS, l'ID dell'VPCendpoint 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'VPCendpoint IDs con valori validi per il tuo account.

```
{
    "Id": "example-key-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM policies",
```

```
            "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-1234abcdf5678c90a"
                }
            }
        }

    ]
}
```

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

Il seguente esempio di policy chiave consente i comandi che gestiscono la KMS chiave solo quando provengono. `vpc-12345678` Inoltre, consente i comandi che utilizzano la KMS chiave per operazioni crittografiche solo quando provengono da`vpc-2b2b2b2b`. È possibile utilizzare una politica come questa se un'applicazione è in esecuzione in una di esseVPC, ma se ne utilizza una seconda, isolata VPC per le funzioni di gestione.

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

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

```
        {
            "Sid": "Allow administrative actions from vpc-12345678",
            "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": "Allow key usage from vpc-2b2b2b2b",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Encrypt","kms:Decrypt","kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "Allow read actions from everywhere",
            "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 VPC endpoint, l'ID dell'VPCendpoint viene visualizzato nella voce di AWS CloudTrail registro che registra la richiesta. È possibile utilizzare l'ID dell'endpoint per verificare l'utilizzo dell'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 KMS chiavi e alias in altri account. Inoltre, per proteggere l'utenteVPC, le richieste respinte da una policy sugli VPC endpoint, ma che altrimenti sarebbero state consentite, non vengono registrate in. AWS CloudTrail

Ad esempio, questa voce di registro di esempio registra una GenerateDataKeyrichiesta che ha utilizzato l'VPCendpoint. 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"
}
```

AWS KMS concetti

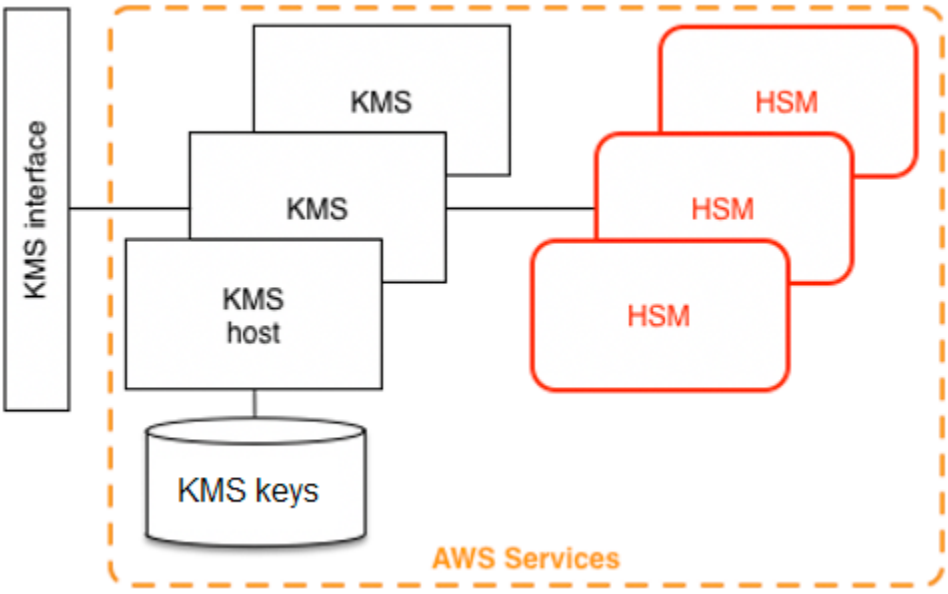
Scopri i termini e i concetti di base utilizzati in AWS Key Management Service (AWS KMS) e come interagiscono per proteggere i tuoi dati.

Introduzione a AWS KMS

AWS Key Management Service (AWS KMS) fornisce un'interfaccia web per generare e gestire chiavi crittografiche e funge da 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 RESTful API operazioni per richiedere operazioni crittografiche di una flotta distribuita di FIPS 140-2 moduli di sicurezza hardware convalidati (). AWS Management Console HSMs AWS KMS HSMSi tratta di un'appliance crittografica hardware standalone multichip progettata per fornire funzioni crittografiche dedicate per soddisfare i requisiti di sicurezza e scalabilità di. AWS KMSÈ possibile stabilire la propria gerarchia crittografica HSM basata su 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 KMS chiavi, ciascuna rappresentata dal relativo ID chiave. Solo con AWS IAM ruoli e account amministrati da ciascun cliente è possibile creare, eliminare o utilizzare KMS le chiavi gestite dal cliente per crittografare, decrittografare, firmare o verificare i dati. È possibile definire i controlli di accesso su chi può gestire e/o utilizzare KMS le chiavi creando una policy allegata alla chiave. Tali politiche 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 mediante 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 KMS chiavi.



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 lo consentono solo TLS 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. AWS API

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 chiavi in testo sempliceKMS. 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. HSMsDispongono di un'ulteriore coppia di chiavi private e pubbliche per stabilire chiavi di crittografia che proteggono la sincronizzazione degli HSM stati.

AWS KMS keys

Le KMS chiavi 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 KMS le chiavi 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 di utilizzo e/o 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. An Chiave gestita

da AWSè una KMS chiave 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 EBS i volumi e solo per i volumi utilizzati dai IAM responsabili dello stesso account della chiave. Pensa a una Chiave gestita da AWS che è 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 AWSper crittografare i dati dei clienti per impostazione predefinita. An Chiave di proprietà di AWS è una KMS chiave presente in un account gestito dal AWS servizio, pertanto gli operatori del servizio hanno 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 aree geografiche senza che l'utente debba preoccuparsi delle autorizzazioni chiave. Utilizzali 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 Chiavi di proprietà di AWS usatele quando la comodità è più importante.

	Chiavi gestite dal cliente	Chiavi gestite da AWS	Chiavi di proprietà di AWS
Policy della chiave	Controllato esclusiva mente dal cliente	Controllato dal servizio; visualizzabile dal cliente	Controllato esclusiva mente 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'elimina zione 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 proporzio nalmente) ogni ora). Addebitat o anche per l'utilizzo delle chiavi	Nessun canone mensile, ma al chiamante viene addebitato API l'utilizz o di questi tasti	Nessun addebito per il cliente

Le KMS chiavi che crei sono chiavi gestite dal cliente. Servizi AWS che utilizzano KMS chiavi per crittografare le risorse di servizio spesso creano chiavi per voi. KMSle chiavi che vengono Servizi AWS create nel tuo AWS account sono Chiavi gestite da AWS. KMSle chiavi che vengono Servizi AWS create in un account di servizio sono Chiavi di proprietà di AWS.

Tipo di KMS chiave	Può visualizz are i metadati KMS chiave	Può gestire la chiave KMS	Usato solo per il mio Account AWS	Rotazione automatica	Prezzi
Chiave gestita dal cliente	Sì	Sì	Sì	Facoltativo.	Canone mensile (proporzi onale a ora) Tariffa per uso

Tipo di KMS chiave	Può visualizzare i metadati KMS chiave	Può gestire la chiave KMS	Usato solo per il mio Account AWS	Rotazione automatica	Prezzi
Chiave gestita da AWS	Sì	No	Sì	Campo 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 KMS le chiavi. 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 chiavi gestite dai clienti. Altri AWS servizi supportano tutti i tipi di KMS chiavi per consentire la facilità di utilizzo Chiave di proprietà di AWS, la visibilità di una Chiave gestita da AWS chiave o il controllo di una chiave gestita dal cliente. 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 KMS chiavi che crei sono chiavi gestite dal cliente. Le chiavi gestite dal cliente sono KMS chiavi Account AWS che crei, possiedi e gestisci. Hai il pieno controllo su queste KMS chiavi, inclusa la definizione e la manutenzione delle relative politiche, IAM politiche e concessioni chiave, l'attivazione e la disabilitazione, la rotazione del materiale crittografico, l'aggiunta di tag, la creazione di alias che fanno riferimento alle chiavi e la pianificazione dell'eliminazione delle KMS chiavi. KMS

Le chiavi gestite dal cliente vengono visualizzate nella pagina chiavi gestite dal cliente della AWS Management Console per AWS KMS. Per identificare in modo definitivo una chiave gestita dal cliente, utilizza l'operazione. DescribeKey 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 AWSsono KMS chiavi 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, an 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 AWSpagina del AWS Management Console 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'DescribeKeyoperazione. 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).
> Chiavi gestite da AWS I nuovi vengono ruotati automaticamente un anno dopo la creazione e successivamente all'incirca ogni anno.
> Chiavi gestite da AWS Le versioni esistenti vengono ruotate automaticamente un anno dopo la loro rotazione più recente e successivamente ogni anno.

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 KMS chiavi in ciascuna regione del vostro account. Tuttavia, se utilizzate per conto di un responsabile del conto, le KMS chiavi vengono conteggiate ai fini delle quote richieste. Per informazioni dettagliate, consultare Quote.

Chiavi di proprietà di AWS

Chiavi di proprietà di AWSsono una raccolta di KMS chiavi possedute e gestite da un AWS servizio per l'utilizzo in più Account AWS lingue. Sebbene non Chiavi di proprietà di AWS siano presenti nel tuo account Account AWS, un AWS servizio può utilizzare un Chiave di proprietà di AWS 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, an 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.

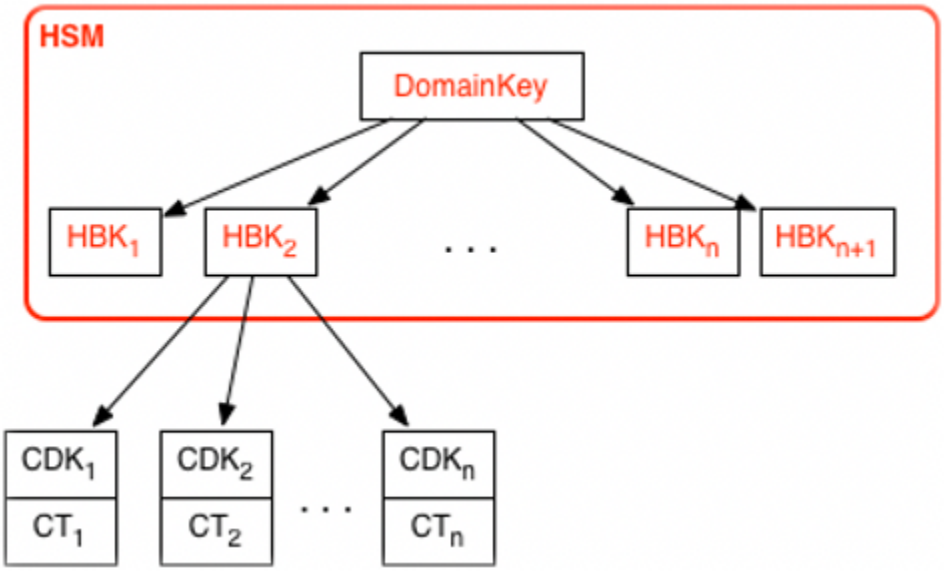
La rotazione di Chiavi di proprietà di AWS varia a seconda dei servizi. Per informazioni sulla rotazione di un determinato servizio Chiave di proprietà di AWS, consulta l'argomento Encryption at Rest nella guida per l'utente o nella guida per sviluppatori del servizio.

AWS KMS key gerarchia

La gerarchia delle chiavi inizia con una chiave logica di primo livello, un. AWS KMS key Una KMS chiave rappresenta un contenitore per materiale chiave di primo livello ed è definita in modo univoco all'interno dello spazio dei nomi del AWS servizio con un Amazon Resource Name (). ARN ARNInclude un identificatore di chiave generato in modo univoco, un ID chiave. Una KMS chiave viene creata sulla base di una richiesta avviata dall'utente tramite. AWS KMS Alla ricezione, AWS KMS richiede la creazione di una chiave di HSM supporto iniziale (HBK) da inserire nel contenitore delle KMS chiavi. HBKViene generato su un HSM dominio ed è progettato per non essere mai esportato dal testo HSM in chiaro. HBKViene invece esportato crittografato con chiavi di dominio HSM -managed. Questi token esportati HBKs sono denominati token chiave esportati (). EKTs

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

All'interno della gerarchia di una KMS chiave specifica, HBK questa può essere considerata come una versione della KMS chiave. Quando si desidera ruotare la KMS chiave AWS KMS, ne HBK viene creata una nuova e associata alla KMS chiave come attiva HBK della chiave. KMS I dati più vecchi HBKs vengono conservati e possono essere utilizzati per decrittografare e verificare dati precedentemente protetti. Ma solo la chiave di crittografia attiva può essere utilizzata per proteggere nuove informazioni.



Puoi richiedere di utilizzare le tue KMS chiavi AWS KMS per proteggere direttamente le informazioni o richiedere chiavi aggiuntive HSM generate che sono protette dalla tua KMS chiave. Queste chiavi

sono chiamate chiavi dei dati del cliente oCDKs. CDKspossono essere restituite crittografate come testo cifrato (CT), in testo non crittografato o entrambi. Tutti gli oggetti crittografati con una KMS chiave (dati forniti dal cliente o chiavi HSM generate) possono essere decrittografati solo su e tramite una chiamata. HSM AWS KMS

Il testo cifrato restituito, o il payload decrittografato, non viene mai archiviato all'interno. AWS KMS Le informazioni vengono restituite all'utente tramite la connessione a. TLS 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	Descrizione	Ciclo di vita
Chiave di dominio	Una GCM chiave a 256 bit AES solo in memoria di una delle versioni HSM utilizzate per avvolgere le KMS chiavi, le chiavi secondarieHSM.	Rotazione giornaliera[1]
HSMchiave di supporto	Una chiave simmetrica a 256 bit RSA o una chiave privata a curva ellittica , utilizzata per proteggere i dati e le chiavi dei clienti e archiviata in modo crittografato sotto chiavi di dominio. Una o più chiavi di HSM supporto comprendono la chiave, rappresen tata da. KMS keyId	Rotazione annuale[2] (config. facoltativa)
Chiave di crittografia derivata	Una chiave a 256 bit AES solo in memoria di una GCM chiave HSM utilizzata per crittografare i dati e le chiavi dei clienti. Derivata da e HBK per ogni crittografia.	Usato una volta per crittografare e rigenerato sulla decrittografia
Chiave dei dati del cliente	Chiave simmetrica o asimmetrica definita dall'utente esportata da testo in chiaro e testo cifrato. HSM	Rotazione e utilizzo controllati dall'applicazione

Chiave	Descrizione	Ciclo di vita
	Crittografato con una chiave di supporto e restituito agli utenti autorizzati tramite canale. HSM TLS	

Di tanto in tanto AWS KMS potrei ridurre la rotazione delle chiavi di dominio portandola al massimo a cadenza settimanale 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 di chiave fungono da nomi per le KMS chiavi. Ti aiutano a riconoscere KMS le tue chiavi nella console. Li usi per indicare quali KMS chiavi desideri utilizzare nelle AWS KMS API operazioni, nelle politiche chiave, nelle politiche e nelle sovvenzioni. IAM I valori degli identificatori chiave sono completamente estranei al materiale chiave associato alla chiave. KMS

AWS KMS definisce diversi identificatori chiave. Quando si crea una KMS chiave, AWS KMS genera una chiave ARN e un ID chiave, che sono proprietà della KMS chiave. Quando crei un alias, AWS KMS genera un alias ARN basato sul nome dell'alias che definisci. È possibile visualizzare gli identificatori di chiave e alias in e in. AWS Management Console AWS KMS API

Nella AWS KMS console, è possibile visualizzare e filtrare KMS le chiavi in base alla chiaveARN, 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 e la chiave della chiave ARN”.

In AWS KMS API, i parametri utilizzati per identificare una KMS chiave sono denominati `KeyId` o varianti, ad esempio o. `TargetKeyId` `DestinationKeyId` Tuttavia, i valori di tali parametri non si limitano alla chiaveIDs. Alcuni possono prendere qualsiasi identificatore di chiave valido. Per informazioni sui valori di ciascun parametro, consultate la descrizione del parametro nel AWS Key Management Service API riferimento.

> **Note**
>
> Quando utilizzate il AWS KMS API, fate attenzione all'identificatore di chiave che utilizzate. 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.

Chiave ARN

La chiave ARN è l'Amazon Resource Name (ARN) di una KMS chiave. È un identificatore unico e completamente qualificato per la KMS chiave. Una chiave ARN include Account AWS la regione e l'ID della chiave. Per informazioni su come trovare la chiave ARN di una KMS chiave, consultathe section called “Trova l'ID e la chiave della chiave ARN”.

Il formato di una chiave ARN è il seguente:

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

Di seguito è riportato un esempio di chiave ARN per una KMS chiave a regione singola.

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

Il *key-id* l'elemento della chiave ARNs delle chiavi multiregionali inizia con il `mrk-` prefisso. Di seguito è riportato un esempio di chiave ARN per una chiave multiregionale.

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

ID chiave

L'ID della chiave identifica in modo univoco una KMS chiave all'interno di un account e di una regione. Per informazioni su come trovare l'ID chiave di una KMS chiave, consulta. the section called “Trova l'ID e la chiave della chiave ARN”

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

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

La chiave IDs 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.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

Alias ARN

L'alias ARN è l'Amazon Resource Name (ARN) di un AWS KMS alias. È un identificatore unico e completo per l'alias e per la KMS chiave che rappresenta. Un alias ARN include la regione e Account AWS il nome dell'alias.

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

Il formato di un alias ARN è il seguente:

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

Quello che segue è l'alias di un ARN fittizio. `ExampleAlias`

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

Nome alias

Il nome alias è una stringa di massimo 256 caratteri. Identifica in modo univoco una KMS chiave associata all'interno di un account e di una regione. In AWS KMS API, i nomi alias iniziano sempre con. `alias/` Per informazioni su come trovare il nome alias di una KMS chiave, 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 asimmetrica rappresenta una KMS chiave pubblica e una coppia di chiavi private 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 asimmetrica, la KMS chiave privata viene creata e non esce mai non crittografata. AWS KMS AWS KMS Per utilizzare la chiave privata, è necessario chiamare. AWS KMSÈ possibile utilizzare la chiave pubblica interna AWS KMS chiamando le AWS KMS API operazioni. 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, KMS le chiavi asimmetriche sono una buona scelta. Tuttavia, se state creando una KMS chiave per crittografare i dati archiviati o gestiti in un AWS servizio, utilizzate una chiave di crittografia simmetrica. KMS AWS i servizi integrati con AWS KMS utilizzano solo KMS chiavi di crittografia simmetriche per crittografare i dati. Questi servizi non supportano la crittografia con chiavi asimmetriche. KMS

AWS KMS supporta tre tipi di chiavi asimmetriche. KMS

RSAKMSchiavi

Una KMS chiave con una coppia di RSA chiavi 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 i dettagli tecnici sugli algoritmi di crittografia e firma che AWS KMS supportano RSA KMS le chiavi, consulta le specifiche RSAchiave.

Tasti Elliptic Curve () ECC KMS

Una KMS chiave 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 ECC KMS chiavi, consulta Specifiche chiave della curva ellittica.

SM2KMSchiavi (solo regioni cinesi)

Una KMS chiave con una coppia di SM2 chiavi per la crittografia e la decrittografia, la firma e la verifica o la derivazione di segreti condivisi (è necessario scegliere un tipo di utilizzo della chiave).

Per dettagli tecnici sugli algoritmi di crittografia e firma che AWS KMS supportano SM2 KMS le chiavi (solo per le regioni cinesi), consulta le specifiche chiave. SM2

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

Regioni

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

Ulteriori informazioni

- Per creare chiavi asimmetriche, vedere. KMS Creare una chiave asimmetrica KMS
- Per creare chiavi asimmetriche multiregionali, vedereKMS. Creazione di chiavi primarie multiregionali
- Per informazioni su come firmare messaggi e verificare le firme con chiavi asimmetriche, consulta Firma digitale con la nuova funzionalità KMS delle chiavi asimmetriche del Security Blog. AWS KMSAWS
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle chiavi asimmetriche, consulta. KMS Deleting asymmetric KMS keys
- Per identificare e visualizzare le chiavi asimmetriche, vedere. KMS Identifica le chiavi asimmetriche KMS

HMACchiavi dentro AWS KMS

Le chiavi Hash Based Message Authentication Code (HMAC) sono KMS chiavi simmetriche utilizzate per generare e verificare all'interno. HMACs AWS KMS Il materiale chiave unico associato a ciascuna HMAC KMS chiave fornisce la chiave segreta richiesta dagli algoritmiHMAC. È possibile utilizzare una HMAC KMS chiave con le `VerifyMac`operazioni `GenerateMac` and per verificare l'integrità e l'autenticità dei dati all'interno. AWS KMS

HMACgli algoritmi combinano una funzione hash crittografica e una chiave segreta condivisa. Accettano un messaggio e una chiave segreta, ad esempio il materiale chiave contenuto in una HMAC KMS chiave, e restituiscono un codice o un tag univoco a dimensione fissa. 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, garantisce HMAC anche l'autenticità; è impossibile generare un HMAC tag identico senza la chiave segreta. HMACsa volte vengono chiamate firme simmetriche, perché funzionano come le firme digitali, ma utilizzano un'unica chiave sia per la firma che per la verifica.

HMACKMSle chiavi e gli HMAC algoritmi utilizzati sono conformi agli standard di AWS KMS settore definiti nel 2104. RFC L' AWS KMS GenerateMacoperazione genera tag standard. HMAC HMACKMSle chiavi vengono generate in moduli di sicurezza AWS KMS hardware certificati nell'ambito del programma di convalida dei moduli crittografici FIPS 140-2 (tranne nelle regioni di Cina (Pechino) e Cina (Ningxia)) e non vengono mai lasciate non crittografate. AWS KMS Per utilizzare una chiave, è necessario chiamare HMACKMS. AWS KMS

È possibile utilizzare HMAC KMS le chiavi per determinare l'autenticità di un messaggio, ad esempio un JSON Web Token (JWT), informazioni tokenizzate sulla carta di credito o una password inviata. Possono anche essere utilizzate come funzioni sicure di derivazione delle chiavi (KDFs), specialmente nelle applicazioni che richiedono chiavi deterministiche.

HMACKMSle chiavi 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 migliori pratiche consigliano di limitare il periodo durante il quale qualsiasi meccanismo di firma, incluso unHMAC, è efficace. 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. HMACi tag non includono un timestamp, ma puoi includerlo nel token o nel messaggio per aiutarti a rilevare quando è il momento di aggiornare il. HMAC

Operazioni crittografiche supportate

HMACKMSle chiavi supportano solo le `GenerateMac`operazioni `VerifyMac`crittografiche. Non è possibile utilizzare HMAC KMS le chiavi per crittografare dati o firmare messaggi

o utilizzare qualsiasi altro tipo di KMS chiave nelle HMAC operazioni. Quando si utilizza l'`GenerateMac`operazione, si fornisce un messaggio di massimo 4.096 byte, una HMAC KMS chiave e un MAC algoritmo compatibile con le specifiche della HMAC chiave e si calcola il tag. `GenerateMac` HMAC Per verificare un HMAC tag, è necessario fornire il HMAC tag e lo stesso messaggio, HMAC KMS chiave e MAC algoritmo `GenerateMac` utilizzati per calcolare il tag originale. HMAC L'`VerifyMac`operazione calcola il HMAC tag e verifica che sia identico al tag fornito. HMAC Se i HMAC tag di input e quelli calcolati non sono identici, la verifica ha esito negativo.

HMACKMSle chiavi non supportano la rotazione automatica delle chiavi e non è possibile creare una HMAC KMS chiave in un archivio chiavi personalizzato.

Se state creando una KMS chiave per crittografare i dati in un AWS servizio, utilizzate una chiave di crittografia simmetrica. Non è possibile utilizzare alcuna chiave. HMAC KMS

Regioni

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

Ulteriori informazioni

- Per creare HMAC KMS chiavi, vediCrea una HMAC KMS chiave.
- Per creare HMAC KMS chiavi multiregionali, vedereChiavi multiregionali in ingresso AWS KMS.
- Per esaminare la differenza nella politica dei tasti predefinita impostata dalla AWS KMS console per HMAC KMS le chiavi, consultathe section called “Consente agli utenti chiave di utilizzare una KMS chiave per operazioni crittografiche”.
- Per identificare e visualizzare HMAC KMS le chiavi, vedereIdentifica le chiavi HMAC KMS.
- Per ulteriori informazioni sull'utilizzo HMACs per creare token JSON Web, consulta How to protect HMACs inside AWS KMS nel AWS Security Blog.
- Ascolta un podcast: Introduzione HMACs a The AWS Key Management Service AWS Official Podcast.

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 si avesse 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 KMS chiavi, le chiavi multiregionali non escono mai non crittografate. AWS KMS È possibile creare chiavi multiregione simmetriche o asimmetriche per la crittografia o la firma, creare chiavi HMAC multiregionali per generare e verificare HMAC tag e creare chiavi multiregione con materiale chiave importato o materiale chiave generato. 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 intermedieCAs, 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 Amazon S3 lato client. AWS Encryption SDK

AWS i servizi che si integrano con AWS KMS la crittografia a riposo o le firme digitali attualmente trattano le chiavi multiregionali come se fossero chiavi a regione singola. Potrebbero riavvolgere o crittografare i dati spostati tra le regioni. Ad esempio, la replica interregionale di Amazon S3 decrittografa e cripta nuovamente i dati con una KMS chiave nella regione di destinazione, anche durante la replica di oggetti protetti da una chiave multiregionale.

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 replica automaticamente AWS KMS chiavi multiregionali in alcuna regione per tuo conto. Chiavi gestite da AWS, le KMS chiavi che AWS i servizi creano per te nell'account sono sempre chiavi per regione singola.

Nelle regioni della Cina, puoi utilizzare la funzionalità chiave multiregionale per replicare KMS le chiavi 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 multiregionali correlate conta come una KMS chiave per i prezzi e le quote.AWS KMS le quote vengono 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 KMS chiavi supportati

È possibile creare i seguenti tipi di KMS chiavi multiregionali:

- Chiavi di crittografia simmetriche KMS
- Chiavi asimmetriche KMS
- HMACKMSchiavi
- KMSchiavi con materiale 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 KMS chiavi multiregionali, consultaControlla l'accesso alle chiavi multiregionali.
- Per creare KMS chiavi primarie multiregionali di qualsiasi tipo, consulta. Creazione di chiavi primarie multiregionali
- Per creare KMS chiavi di replica multiregionali, vedere. Creazione di 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 KMS chiavi multiregionali, vedereIdentifica le chiavi HMAC KMS.
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle chiavi multiregionaliKMS, 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 multiregionale fa parte di un set di KMS chiavi con lo stesso ID chiave e lo stesso materiale chiave (e altre proprietà condivise) ma diversi. Regioni AWS Ogni chiave multiregionale è una chiave perfettamente funzionante che può essere utilizzata in modo completamente indipendente dalle relative KMS chiavi multiregionali. Poiché tutte le chiavi multiregionali correlate hanno lo stesso ID di chiave e lo stesso materiale chiave, sono interoperabili, ovvero qualsiasi chiave multiregionale correlata Regione AWS può decrittografare il testo cifrato crittografato da qualsiasi altra chiave multiregionale correlata.

La proprietà multiregionale di una chiave viene impostata al momento della creazione. KMS 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 AWS KMS materiale chiave o materiale chiave importato. 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 altri 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 KMS chiave 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 utilizzarla come qualsiasi altra KMS chiave e replicarla 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 KMS chiave che ha lo stesso ID e lo stesso materiale chiave della chiave primaria e delle relative chiavi di replica, ma esiste in un altro. Regione AWS

Una chiave di replica è una chiave completamente funzionale con criteri KMS chiave, concessioni, alias, tag e altre proprietà propri. 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 Regione AWS nella stessa partizione. 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'`Region`elemento della chiave è ARN diverso).
- 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, puoi garantire la residenza dei dati e la conformità alla sovranità dei dati. KMSle chiavi in una determinata regione possono decrittografare i dati sensibili solo in quella regione. I dati crittografati in una regione possono rimanere completamente protetti e inaccessibili in qualsiasi altra regione.

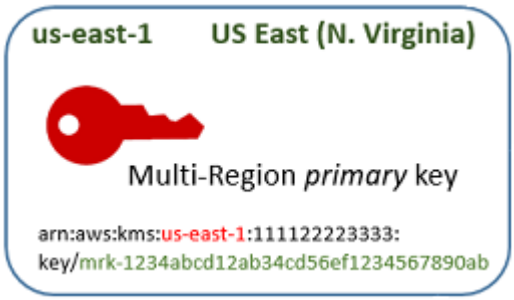
  Per verificare la residenza e la sovranità dei dati con chiavi multiregionali, è necessario implementare politiche di accesso e compilare eventi in più regioni. AWS CloudTrail

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

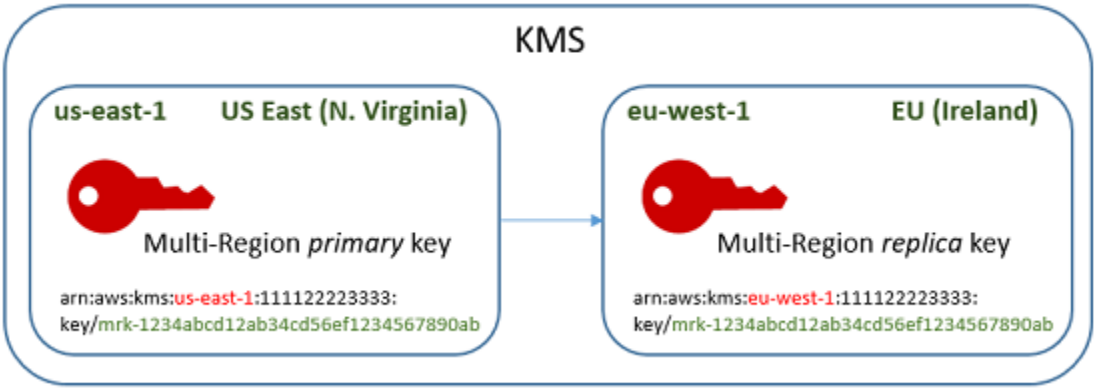
Si inizia creando una chiave primaria multiregionale simmetrica o asimmetrica in un sistema che supporti, ad esempio Stati Uniti orientali (Virginia settentrionale). Regione AWS AWS KMS È possibile decidere se una chiave è di regione singola o multi-regione solo al momento della creazione. Non è possibile modificare questa proprietà in un secondo momento. Come per qualsiasi KMS chiave, si imposta una politica chiave per la chiave multiregionale e si possono creare concessioni e aggiungere alias e tag per la categorizzazione e l'autorizzazione. (Queste sono proprietà indipendenti che non sono condiviei 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 il CreateKeyAPIparametro impostato su. `MultiRegion` `true` 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. Quindi trasporta in modo sicuro il materiale chiave attraverso il confine della regione e lo associa alla nuova KMS chiave nella regione di destinazione, il tutto all'interno. 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 console o utilizzando. AWS KMS ReplicateKeyAPI



La chiave di replica multiregionale risultante è una chiave completamente funzionale con le stesse proprietà condivise della KMS chiave primaria. Sotto tutti gli altri aspetti, è una KMS chiave indipendente con una descrizione, una politica chiave, 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.

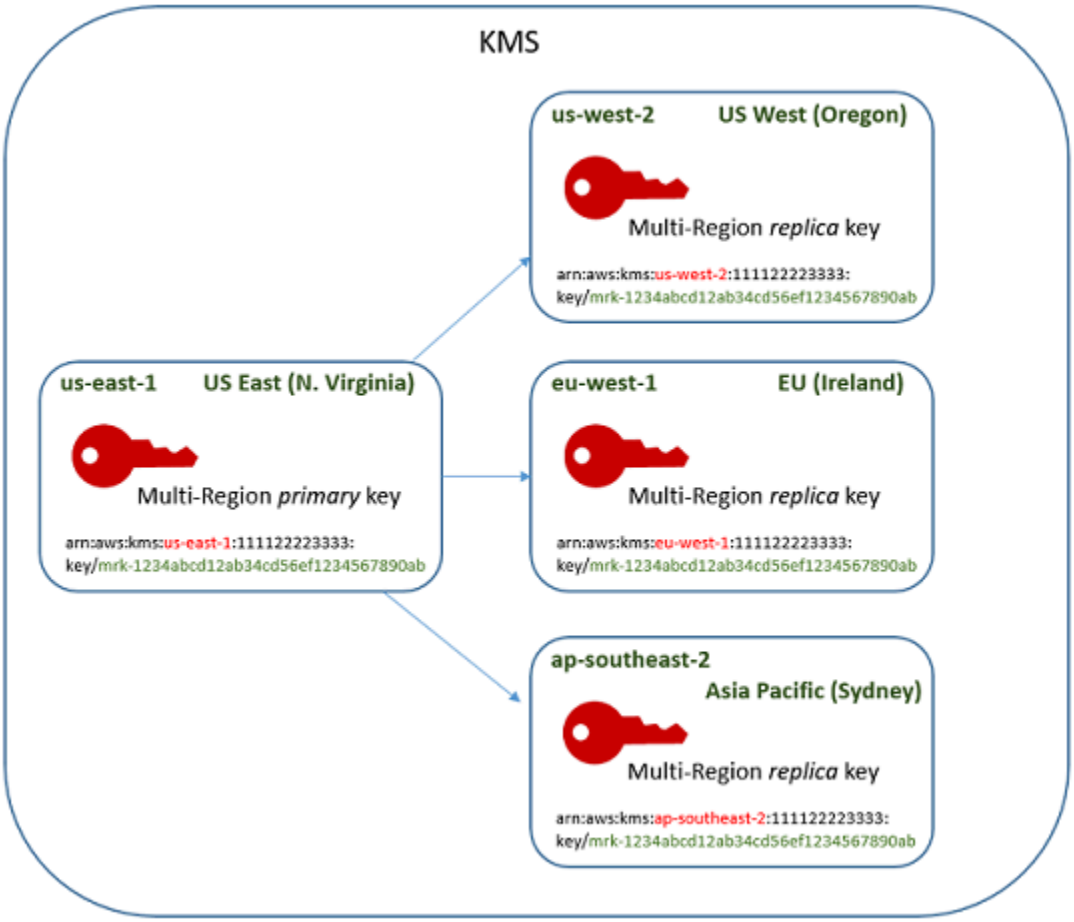
Le chiavi multi-regione correlate hanno lo stesso ID chiave. La loro chiave ARNs (Amazon Resource Names) differisce solo nel campo Regione. Ad esempio, la chiave primaria multiregionale e le

chiavi di replica potrebbero avere la seguente chiave di esempio. ARNs L'ID della chiave, l'ultimo elemento della chiave, è ARN identico. Entrambe le chiavi hanno l'ID chiave distintivo delle chiavi multiregionali, 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 KMS chiave al testo cifrato in modo che il testo cifrato possa essere decrittografato solo con quella KMS chiave o con una chiave con lo stesso ID di chiave. KMS 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, specifica la regione o distinguila. ARN

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 chiaveIDs, 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'SynchronizeMultiRegionKeyevento 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.

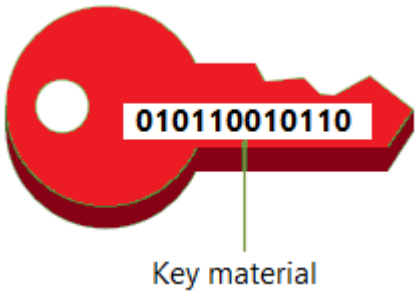
Modifica della chiave primaria — Ogni set di chiavi multi-regione deve avere esattamente una chiave primaria. La chiave primaria è l'unica chiave che può essere replicata. È anche la fonte delle proprietà condivise delle chiavi di replica. Tuttavia, è possibile modificare la chiave primaria in una replica 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 KMS chiavi, 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, consultare Deleting multi-Region keys.

Importazione di materiale chiave per le AWS KMS chiavi

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

Una KMS chiave è una rappresentazione logica di una chiave di dati. I metadati di una KMS chiave includono l'ID del materiale chiave utilizzato per eseguire operazioni crittografiche. Quando si crea una KMS chiave, per impostazione predefinita, AWS KMS genera il materiale chiave per quella KMS chiave. Tuttavia, è possibile creare una KMS chiave senza materiale chiave e quindi importare il proprio materiale chiave in quella KMS chiave, una funzionalità spesso nota come «porta la propria chiave» (BYOK).



> **Note**
>
> AWS KMS non supporta la decrittografia di alcun AWS KMS testo cifrato crittografato con una chiave di crittografia simmetrica esterna a AWS KMS, anche se il testo cifrato è stato crittografato con una KMS chiave con materiale chiave importato. KMS 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 chiavi per la firma del codice AWS KMS, la firma dei PKI certificati e le applicazioni bloccate con certificati
- Per impostare una scadenza per il materiale chiave AWS e per eliminarlo manualmente, ma anche per renderlo nuovamente disponibile in futuro. Al contrario, la pianificazione dell'eliminazione delle chiavi richiede un periodo di attesa compreso tra 7 e 30 giorni, dopo il quale non è possibile recuperare la KMS chiave 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 chiavi e chiavi asimmetriche, l'importazione crea HMAC chiavi compatibili e interoperabili che funzionano all'interno e all'esterno di. AWS

Tipi di chiavi supportati KMS

AWS KMS supporta materiale chiave importato per i seguenti tipi di KMS chiavi. Non è possibile importare materiale chiave nelle KMS chiavi degli archivi di chiavi personalizzati.

- Chiavi di crittografia simmetriche KMS
- Chiavi asimmetriche KMS
- HMACKMSchiavi
- Chiavi multi-regione di tutti i tipi supportati.

Regioni

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

Nelle regioni della Cina, i requisiti materiali chiave per le KMS chiavi di crittografia simmetrica sono diversi da quelli delle altre regioni. Per informazioni dettagliate, consultare Fase 3: crittografare il materiale delle chiavi.

Ulteriori informazioni

- Per creare KMS chiavi con materiale chiave importato, vedere. Crea una KMS chiave con materiale chiave importato
- Per creare un allarme che avvisi quando il materiale chiave importato in una KMS chiave sta per scadere, vedi. Crea un CloudWatch allarme per la scadenza del materiale chiave importato
- Per reimportare il materiale chiave in una KMS chiave, vedere. Reimporta il materiale chiave
- Per identificare e visualizzare KMS le chiavi con materiale chiave importato, vedereIdentifica KMS le chiavi con materiale chiave importato.
- Per ulteriori informazioni sulle considerazioni speciali relative all'eliminazione di KMS chiavi con materiale chiave importato, vedere. 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.

Puoi eliminare il materiale chiave

È possibile eliminare il materiale chiave importato da una KMS chiave, rendendola immediatamente inutilizzabile. KMS Inoltre, quando importate materiale chiave in una KMS chiave, potete determinare se la chiave scade e impostarne l'ora di scadenza. Quando arriva la data di scadenza, AWS KMS elimina il materiale chiave. Senza materiale chiave, la KMS chiave non può essere utilizzata in alcuna operazione crittografica. Per ripristinare la chiave, è necessario importare nuovamente lo stesso materiale chiave nella chiave.

Non puoi modificare il materiale della chiave

Quando importate il materiale chiave in una KMS chiave, la KMS chiave viene associata in modo permanente a quel materiale chiave. È possibile reimportare lo stesso materiale chiave, ma non è

possibile importare materiale chiave diverso in quella KMS chiave. Inoltre, non è possibile abilitare la rotazione automatica dei tasti per una KMS chiave con materiale chiave importato. Tuttavia, è possibile ruotare manualmente una KMS chiave con materiale chiave importato.

Non puoi modificare l'origine del materiale della chiave

KMSle chiavi progettate per il materiale chiave importato hanno un valore di origine `EXTERNAL` che non può essere modificato. Non è possibile convertire una KMS chiave 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 KMS chiave con materiale AWS KMS chiave in una chiave progettata per 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 aree multiregionali con materiale chiave importato hanno le stesse caratteristiche delle KMS chiavi con materiale chiave importato e possono interagire tra loro. Regioni AWS Per creare una chiave multiregionale con materiale chiave importato, è necessario importare lo stesso materiale chiave nella chiave primaria e in ogni KMS chiave di replica.

Le chiavi e le chiavi asimmetriche sono portatili e interoperabili HMAC

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

A differenza del testo cifrato AWS KMS simmetrico, che è indissolubilmente legato alla KMS chiave utilizzata nell'algoritmo, AWS KMS utilizza formati standard HMAC 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 KMS chiave ha importato materiale chiave, è possibile utilizzare il materiale chiave importato all'esterno di per eseguire le seguenti operazioni. AWS

- HMACchiavi: è possibile verificare un HMAC tag generato dalla HMAC KMS chiave con materiale chiave importato. È inoltre possibile utilizzare la HMAC KMS chiave con il materiale chiave importato per verificare un HMAC tag 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 con la chiave pubblica corrispondente. KMS Puoi anche usare la tua chiave asimmetrica per decrittografare un testo cifrato asimmetrico generato KMS all'esterno di. AWS
- Chiavi di firma asimmetriche: puoi utilizzare la tua chiave di firma asimmetrica con materiale chiave importato per verificare le firme digitali generate dalla tua KMS 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 asimmetrica. AWS KMS
- Chiavi contrattuali a chiave asimmetrica: puoi utilizzare la tua chiave di accordo a chiave asimmetrica con materiale chiave importato per ricavare segreti condivisi con un peer KMS esterno. AWS

Se importi lo stesso materiale chiave in chiavi diverse all'interno della stessa chiave, anche quelle KMS chiavi sono interoperabili. Regione AWS Per creare KMS chiavi interoperabili in diversi formati Regioni AWS, create una chiave multiregionale con materiale chiave importato.

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 con materiale chiave AWS KMS importato. KMS
- KMSle chiavi con lo stesso materiale chiave importato non sono interoperabili. Il testo cifrato simmetrico che AWS KMS genera testo cifrato specifico per ogni chiave. KMS Questo formato di testo cifrato garantisce che solo la chiave contenente i dati crittografati possa decrittografarliKMS.

Inoltre, non è possibile utilizzare AWS strumenti, come la AWS Encryption SDKcrittografia lato client di 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.

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, consultare Protezione del materiale della chiave importato.

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 RSA coppia di chiavi generata nei moduli di sicurezza AWS KMS hardware (HSMs) convalidati nell'ambito del FIPS140-2 Cryptographic Module Validation Program. È possibile crittografare il materiale chiave direttamente con la chiave pubblica di avvolgimento oppure crittografare il materiale chiave con una chiave simmetrica e quindi crittografare la chiave AES simmetrica con la chiave pubblica. AES RSA

Al ricevimento, AWS KMS decripta il materiale chiave con la chiave privata corrispondente in un formato AWS KMS HSM e lo cripta nuovamente con una chiave simmetrica che esiste solo nella memoria volatile diAES. HSM Il materiale chiave non viene mai lasciato in formato testo normale. HSM Viene decifrato solo mentre è in uso e solo all'interno. AWS KMS HSMs

L'uso della KMS chiave con materiale chiave importato è determinato esclusivamente dalle politiche di controllo dell'accesso impostate sulla KMS chiave. Inoltre, è possibile utilizzare alias e tag per identificare e controllare l'accesso alla KMS chiave. Puoi 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 KMS chiave 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 è possibile eliminare il materiale chiave AWS KMS generato per una KMS chiave e non è possibile impostare la scadenza del materiale AWS KMS chiave, sebbene sia possibile ruotarlo.

- Quando eliminate manualmente il materiale chiave importato, AWS KMS elimina il materiale chiave ma non elimina la KMS chiave o i relativi metadati. Al contrario, la pianificazione dell'eliminazione delle chiavi richiede un periodo di attesa compreso tra 7 e 30 giorni, dopodiché elimina AWS KMS definitivamente la KMS chiave, 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 KMS chiave e i relativi metadati.

È necessario conservare una copia del materiale chiave importato all'esterno AWS di un sistema controllato dall'utente. Si consiglia di archiviare una copia esportabile del materiale chiave importato in un sistema di gestione delle chiavi, ad esempio unHSM. Se il materiale chiave importato viene eliminato o scade, la KMS chiave associata diventa inutilizzabile finché non si reimporta lo stesso materiale chiave. Se il materiale chiave importato viene perso definitivamente, qualsiasi testo cifrato crittografato sotto la chiave è irrecuperabile. KMS

KMSchiavi in un archivio di HSM chiavi Cloud

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

Funzionalità supportate

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

- Utilizza le politiche, le IAM politiche e le concessioni chiave per autorizzare l'accesso alle chiavi. KMS

- Abilita e disabilita i tasti. KMS
- Assegna tag e crea alias e usa il controllo di accesso basato sugli attributi (ABAC) per autorizzare l'accesso alle chiavi. KMS
- Utilizzate le KMS chiavi per eseguire le seguenti operazioni crittografiche:
  - Encrypt
  - Decrypt
  - GenerateDataKey
  - GenerateDataKeyWithoutPlaintext
  - ReEncrypt

  Le operazioni che generano coppie di chiavi di dati asimmetriche GenerateDataKeyPaire GenerateDataKeyPairWithoutPlaintext, non sono supportate negli archivi di chiavi personalizzati.
- Utilizza le KMS chiavi con AWS servizi che si integrano AWS KMS e supportano le chiavi gestite dai clienti.
- Tieni traccia dell'uso delle tue KMS chiavi nei AWS CloudTrail log e negli strumenti di CloudWatch monitoraggio di Amazon.

Caratteristiche non supportate

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

Utilizzo KMS delle chiavi in un archivio di AWS CloudHSM chiavi

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

Tuttavia, quando si utilizza una KMS chiave in un archivio di AWS CloudHSM chiavi, l'operazione di crittografia viene eseguita interamente all'interno del AWS CloudHSM cluster associato all'

AWS CloudHSM archivio chiavi. L'operazione utilizza il materiale chiave del cluster associato alla KMS chiave scelta.

Perché ciò avvenga, devono essere soddisfatte le seguenti condizioni.

- Lo stato chiave della KMS chiave deve essere`Enabled`. Per trovare lo stato della chiave, usa il campo Status nella AWS KMS console o il `KeyState` campo nella DescribeKeyrisposta.
- L'archivio delle AWS CloudHSM chiavi deve essere connesso al relativo AWS CloudHSM cluster. Il relativo stato nella AWS KMS console o `ConnectionState` nella DescribeCustomKeyStoresrisposta deve essere`CONNECTED`.
- Il AWS CloudHSM cluster associato all'archivio chiavi personalizzato deve contenerne almeno uno attivoHSM. Per trovare il numero di persone attive HSMs nel cluster, usa la AWS KMS console, la AWS CloudHSM console o l'DescribeClustersoperazione.
- Il AWS CloudHSM cluster deve contenere il materiale chiave per la KMS chiave. Se il materiale chiave è stato eliminato dal cluster o ne HSM è stato creato uno da un backup che non includeva il materiale chiave, l'operazione di crittografia avrà esito negativo.

Se queste condizioni non vengono 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 difettosa KMS.

Quando utilizzate le KMS chiavi in un archivio di AWS CloudHSM chiavi, tenete presente che le KMS chiavi di ogni archivio di AWS CloudHSM chiavi condividono una quota di richiesta di archivio chiavi personalizzata per le operazioni crittografiche. 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, consultaAWS CloudHSM negozi chiave.
- Per creare KMS chiavi in un archivio di AWS CloudHSM chiavi, consultaCreare una KMS chiave in un archivio di AWS CloudHSM chiavi.

- Per identificare e visualizzare KMS le chiavi in un archivio di AWS CloudHSM chiavi, vedereIdentifica KMS le chiavi negli archivi AWS CloudHSM delle chiavi.
- Per trovare KMS chiavi e materiale chiave in un archivio di AWS CloudHSM chiavi, vediTrova KMS chiavi e materiale chiave in un negozio di AWS CloudHSM chiavi.
- Per informazioni su considerazioni speciali sull'eliminazione delle KMS chiavi in un archivio AWS CloudHSM chiavi, vedi Eliminazione delle KMS chiavi da un AWS CloudHSM archivio chiavi.

KMSchiavi in archivi di chiavi esterni

Per creare, visualizzare, gestire, utilizzare e pianificare l'eliminazione delle KMS chiavi in un archivio di chiavi esterno, si utilizzano procedure molto simili a quelle utilizzate per KMS le altre chiavi. Tuttavia, quando si crea una KMS chiave in un archivio di chiavi esterno, si specificano un archivio chiavi esterno e una chiave esterna. Quando si utilizza una KMS chiave in un archivio di chiavi esterno, le operazioni di crittografia e decrittografia vengono eseguite dal gestore delle chiavi esterno utilizzando la chiave esterna specificata.

AWS KMS non è possibile 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 KMS chiave in un archivio di chiavi esterno, l'archivio di chiavi esterno che ospita la KMS chiave deve essere collegato al relativo proxy di archiviazione chiavi esterno.

Funzionalità supportate

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

- Utilizza le politiche, IAMle politiche e le concessioni chiave per controllare l'accesso alle KMS chiavi.
- Abilita e disabilita i KMS tasti. Queste azioni non influiscono sulla chiave esterna nel gestore delle chiavi esterne.
- Assegna tag e crea alias e usa il controllo di accesso basato sugli attributi (ABAC) per autorizzare l'accesso alle chiavi. KMS
- Utilizzate le KMS chiavi per eseguire le seguenti operazioni crittografiche:
  - Encrypt

- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Le operazioni che generano coppie di chiavi di dati asimmetriche GenerateDataKeyPaire GenerateDataKeyPairWithoutPlaintext, non sono supportate negli archivi di chiavi personalizzati.

- Utilizza le KMS chiavi Servizi AWS che si integrano AWS KMS e supportano le chiavi gestite dai clienti.

Caratteristiche non supportate

- Gli archivi di chiavi esterni supportano solo chiavi di crittografia KMS simmetriche. Non è possibile creare HMAC KMS chiavi o chiavi asimmetriche in un archivio di KMS chiavi esterno.
- GenerateDataKeyPaire non GenerateDataKeyPairWithoutPlaintextsono supportati sulle KMS chiavi in un archivio di chiavi esterno.
- Non è possibile utilizzare un AWS CloudFormation modelloAWS:KMS:: :Key per creare un archivio di chiavi esterno o una KMS chiave in un archivio di chiavi esterno.
- Le chiavi multi-regione non sono supportate in un archivio delle chiavi esterne.
- KMSle chiavi con materiale chiave importato non sono supportate in un archivio di chiavi esterno.
- La rotazione automatica delle chiavi non è supportata per KMS le chiavi in un archivio chiavi esterno.

Utilizzo KMS delle chiavi in un archivio di chiavi esterno

Quando utilizzi la KMS chiave in una richiesta, identifica la KMS chiave tramite il relativo ID chiave, chiaveARN, alias o ARN alias. Non è necessario specificare l'archivio delle chiavi esterne. La risposta include gli stessi campi restituiti per qualsiasi chiave di crittografia simmetrica. KMS Tuttavia, quando si utilizza una KMS chiave in un archivio di chiavi esterno, le operazioni di crittografia e decrittografia vengono eseguite dal gestore delle chiavi esterno utilizzando la chiave esterna associata alla chiave. KMS

Per garantire che il testo cifrato crittografato da una KMS chiave in un archivio di chiavi esterno sia sicuro almeno quanto qualsiasi testo cifrato crittografato con una chiave standardKMS, utilizza la doppia crittografia. AWS KMS I dati vengono prima crittografati utilizzando il materiale chiave AWS KMS . AWS KMS Quindi vengono crittografati dal gestore delle chiavi esterno utilizzando la

chiave esterna della KMS chiave. Per decrittografare il testo cifrato a doppia crittografia, il testo cifrato viene prima decrittografato dal gestore delle chiavi esterno utilizzando la chiave esterna per la chiave. KMS Quindi viene decrittografato utilizzando il materiale chiave per la chiave. AWS KMS AWS KMS KMS

Perché ciò avvenga, devono essere soddisfatte le seguenti condizioni.

- Lo stato chiave della KMS chiave deve essere. `Enabled` Per trovare lo stato della chiave, consulta il campo Stato relativo alle chiavi gestite dal cliente, la AWS KMS console o il `KeyState` campo nella DescribeKeyrisposta.
- L'archivio chiavi esterno che ospita la KMS chiave deve essere connesso al relativo proxy di archiviazione chiavi esterno, ovvero lo stato di connessione dell'archivio chiavi esterno deve essere`CONNECTED`.

  È possibile visualizzare lo stato della connessione nella pagina Archivi di chiavi esterni nella AWS KMS console o nella DescribeCustomKeyStoresrisposta. Lo stato di connessione dell'archivio chiavi esterno viene visualizzato anche nella pagina di dettaglio della KMS chiave nella AWS KMS console. 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, consulta 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 influenzato dalle modifiche allo stato della KMS chiave, incluse l'attivazione e la disabilitazione della KMS chiave. Analogamente, la disabilitazione o l'eliminazione della chiave esterna non modifica lo stato della KMS chiave, ma le operazioni di crittografia che utilizzano la chiave associata KMS falliranno.

Se queste condizioni non vengono soddisfatte, l'operazione di crittografia ha esito negativo e AWS KMS restituisce un'eccezione. `KMSInvalidStateException` 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 utilizzate le KMS chiavi in un archivio di chiavi esterno, tenete presente che le KMS chiavi in ogni archivio di chiavi esterno condividono una quota di richiesta di archiviazione chiavi personalizzata per le operazioni di crittografia. Se superi la quota, AWS KMS restituisce un`ThrottlingException`. 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, consultaArchivi delle chiavi esterne.
- Per ulteriori informazioni sul materiale chiave negli archivi di chiavi esterni, consultaChiave esterna.
- Per creare KMS chiavi in un archivio di chiavi esterno, vedereCreare una KMS chiave in archivi di chiavi esterni.
- Per identificare e visualizzare KMS le chiavi in un archivio di chiavi esterno, vedereIdentifica KMS le chiavi negli archivi di chiavi esterni.
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle KMS chiavi in un archivio di chiavi esterno, vedere Eliminazione delle KMS chiavi 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 iniziale predefinito di algoritmi crittografici è stato selezionato tra gli algoritmi del Federal Information Processing Standard (FIPS-approved) per le relative proprietà di sicurezza e prestazioni.

Entropia e generazione di numeri casuali

AWS KMS la generazione delle chiavi viene eseguita in. AWS KMS HSMs HSMsImplementano 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) utilizzano chiavi a 256 bit. Le chiamate analoghe per decrittografare utilizzano la funzione inversa.

AES- GCM è uno schema di crittografia autenticato. Oltre a crittografare il testo non crittografato per produrre testo cifrato, calcola un tag di autenticazione sul testo cifrato e su tutti i dati aggiuntivi per i quali è richiesta l'autenticazione (dati autenticati aggiuntivi o). AAD Il tag di autenticazione aiuta a garantire che i dati provengano dalla presunta fonte e che il testo cifrato non sia stato modificato. AAD

Spesso, AWS omette l'inclusione di 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 suddivisa tra il testo in chiaro da crittografare e il testo in chiaro da proteggere. AAD

AWS KMS offre la possibilità di importare il materiale chiave in un AWS KMS key anziché fare affidamento su di esso per generare il materiale chiave. AWS KMS Questo materiale chiave importato può essere crittografato utilizzando RSAES- OAEP per proteggere la chiave durante il trasporto verso. AWS KMS HSM Le coppie di RSA chiavi vengono generate su AWS KMS HSMs. Il materiale chiave importato viene decrittografato AWS KMS HSM e ricrittografato sottoAES, 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 chiamando le AWS KMS API operazioni, oppure scaricare la chiave pubblica e usarla all'esterno. AWS KMS

AWS KMS supporta i seguenti codici asimmetrici.

- RSA- OAEP (per la crittografia) & RSA - PSS e - RSA PKCS - #1 -v1_5 (per la firma e la verifica) — Supporta le lunghezze delle RSA chiavi (in bit): 2048, 3072 e 4096 per diversi requisiti di sicurezza.
- Elliptic Curve (ECC): utilizzata per la firma e la verifica o la derivazione di segreti condivisi, ma non per entrambi. Supporta ECC le curve: NIST P256, P384, P521, 256k1. SECP

- 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 l'utilizzo di una sola chiave. Supporti SM2PKE per la crittografia e la firma. SM2DSA

Funzioni di derivazione chiave

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

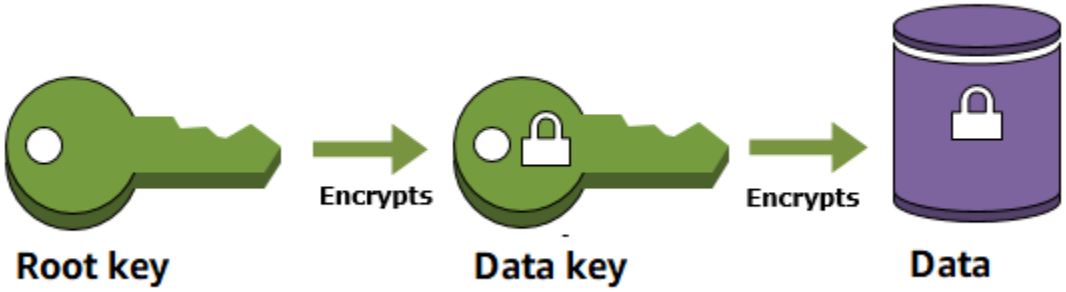
AWS KMS uso interno delle firme digitali

Le firme digitali vengono utilizzate anche per autenticare comandi e comunicazioni tra entità AWS KMS . Tutte le entità di servizio dispongono di una coppia di chiavi dell'algoritmo di firma digitale a curva ellittica (ECDSA). Funzionano ECDSA come definito in Use of Elliptic Curve Cryptography (ECC) Algoritmi in Cryptographic Message Syntax (CMS) e X9.62-2005: Crittografia a chiave pubblica per il settore dei servizi finanziari: The Elliptic Curve Digital Signature Algorithm (). ECDSA Le entità utilizzano l'algoritmo hash sicuro definito nelle Federal Information Processing Standards Publications, 180-4, noto come. FIPS PUB SHA384 Le chiavi vengono generate sulla curva secp384r1 (-P384). NIST

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.



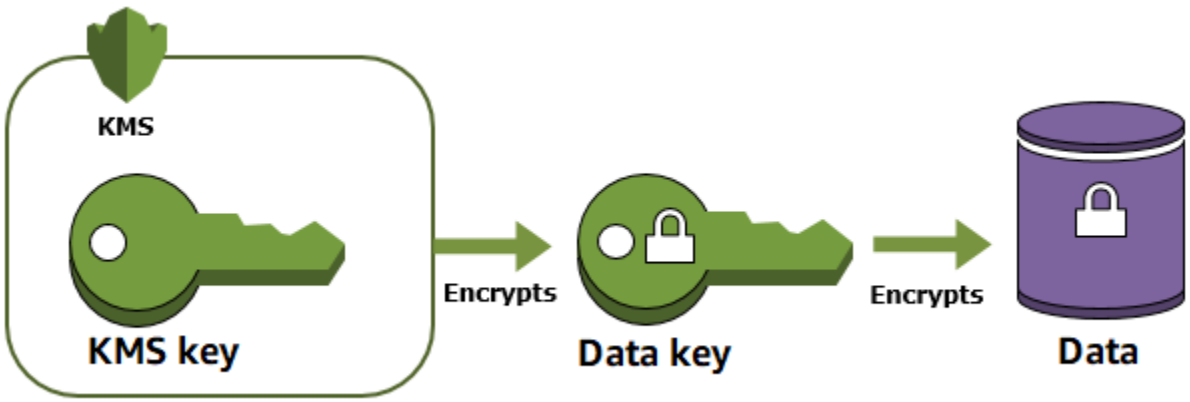
AWS KMS ti aiuta a proteggere le tue chiavi di crittografia archiviandole e gestendole in modo sicuro. La chiave principale memorizzata in AWS KMS, nota come AWS KMS keys, non lascia mai i moduli di sicurezza hardware AWS KMS FIPS convalidati non crittografati. Per utilizzare una KMS chiave, è necessario 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 deriva da una chiave statica a lungo termine k e un'altra chiave è una chiave per messaggio msgKey, generata per crittografare il messaggio. La busta viene formata crittografando il messaggio: ciphertext = Encrypt (, message). msgKey Quindi la chiave del messaggio viene crittografata con la chiave statica a lungo termine: encKey = Encrypt (k,). msgKey Infine, i due valori (encKeytesto cifrato) vengono impacchettati in un'unica struttura, o busta, in un messaggio crittografato.

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 offre 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 archiviarla, 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

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

Per eseguire operazioni crittografiche con KMS le chiavi, utilizzate AWS SDKs, AWS Command Line Interface (AWS CLI) o. AWS Tools for 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. Mostra anche il tipo di chiave e i requisiti di utilizzo KMS delle chiavi utilizzate nell'operazione.

Operazione	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 supportato sulle KMS chiavi negli archivi di chiavi personalizzati.	`ENCRYPT_DECRYPT`
GenerateDataKeyPairWithoutPlaintext	Simmetrica [1] Non supportato sulle KMS chiavi negli archivi di chiavi personalizzati.	`ENCRYPT_DECRYPT`
GenerateDataKeyWithoutPlaintext	Simmetria	`ENCRYPT_DECRYPT`
GenerateMac	HMAC	`GENERATE_VERIFY_MAC`
GenerateRandom	N/A. Questa operazione non	N/D

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

[1] Genera una coppia di chiavi dati asimmetrica protetta da una chiave di crittografia simmetrica. KMS

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 al secondo. Per informazioni dettagliate, consultare the section called “Quote condivise per le operazioni di crittografia”.

KMSaccesso con chiavi e autorizzazioni

Per utilizzarlo AWS KMS, è necessario disporre di credenziali che AWS possano essere utilizzate per autenticare le 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 KMS chiave a meno che tale autorizzazione non sia fornita esplicitamente e mai negata. Non esistono autorizzazioni implicite o automatiche per utilizzare o gestire una chiave. KMS

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

- Politica chiave: ogni KMS chiave ha una politica chiave. È il meccanismo principale per controllare l'accesso a una KMS chiave. È possibile utilizzare solo la politica chiave per controllare l'accesso, il che significa che l'intero ambito di accesso alla KMS chiave è definito in un unico documento (la politica chiave). Per ulteriori informazioni sull'utilizzo delle policy delle chiavi, consulta Policy delle chiavi.
- IAMpolitiche: è possibile utilizzare IAM le politiche in combinazione con la politica chiave e le concessioni per controllare l'accesso a una KMS chiave. Il controllo dell'accesso eseguito in questo modo consente di gestire tutte le autorizzazioni delle identità IAM in IAM. Per utilizzare una IAM policy che consenta l'accesso a una KMS chiave, la policy chiave deve consentirlo esplicitamente. Per ulteriori informazioni sull'utilizzo delle policy IAM, consulta Policy IAM.
- Sovvenzioni: è possibile utilizzare le sovvenzioni in combinazione con la politica e IAM le politiche chiave per consentire l'accesso a una chiave. KMS Il controllo dell'accesso in questo modo consente di consentire l'accesso alla KMS chiave contenuta nella policy chiave e di consentire alle identità di delegare il proprio accesso ad altri. Per ulteriori informazioni sull'utilizzo di concessioni, consulta Sovvenzioni in AWS KMS.

KMSpolitiche chiave

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

Tutte KMS le chiavi hanno una politica chiave. Se non ne fornisci una, ne AWS KMS crea una per te. La politica delle chiavi predefinita AWS KMS utilizzata varia a seconda che si crei la chiave nella

AWS KMS console o si utilizzi la AWS KMS API. Ti consigliamo di modificare la politica delle chiavi predefinita per allinearla ai requisiti dell'organizzazione per le autorizzazioni con privilegi minimi.

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

È inoltre possibile utilizzare IAM le politiche in combinazione con le politiche e le concessioni chiave per controllare l'accesso a una chiave. KMS Per utilizzare una IAM politica per controllare l'accesso a una KMS chiave, la politica chiave deve concedere all'account l'autorizzazione a utilizzare IAM le politiche. È possibile specificare una dichiarazione politica chiave che abiliti IAM le politiche oppure specificare esplicitamente i principi consentiti nella politica chiave.

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

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

KMSsovvenzioni chiave

Oltre alle IAM politiche chiave, AWS KMS sostiene le sovvenzioni. Le sovvenzioni offrono un modo flessibile e potente per delegare le autorizzazioni. Puoi utilizzare le sovvenzioni per concedere un accesso con KMS chiave limitata nel tempo ai IAM responsabili del tuo AWS account o di altri account. AWS Ti consigliamo di emettere un accesso limitato nel tempo se non conosci i nomi dei responsabili al momento della creazione delle politiche o se i principali che richiedono l'accesso cambiano frequentemente. Il titolare del beneficiario può essere intestato allo stesso conto della chiave o a un conto diverso. KMS Se il principale e la KMS chiave si trovano in conti diversi, è necessario specificare una IAM politica oltre alla sovvenzione. Le sovvenzioni richiedono una gestione aggiuntiva perché è necessario API chiamare un addetto per creare la sovvenzione e ritirarla o revocarla quando non è più necessaria.

I seguenti argomenti forniscono dettagli su come utilizzare AWS Identity and Access Management (IAM) e AWS KMS le autorizzazioni per proteggere le risorse controllando chi può accedervi.

Politiche chiave in AWS KMS

Una politica chiave è una politica delle risorse per un. AWS KMS key Le politiche chiave sono il modo principale per controllare l'accesso alle KMS chiavi. Ogni KMS chiave deve avere esattamente una politica chiave. Le dichiarazioni contenute nella politica chiave determinano chi ha il permesso di usare la KMS chiave e come può usarla. È inoltre possibile utilizzare IAMpolitiche e concessioni per controllare l'accesso alla KMS chiave, ma ogni KMS chiave deve avere una politica chiave.

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

A meno che la politica chiave non lo consenta esplicitamente, non è possibile utilizzare i IAM criteri per consentire l'accesso a una chiave. KMS Senza l'autorizzazione della politica chiave, IAM le politiche che consentono le autorizzazioni non hanno alcun effetto. (È possibile utilizzare una IAM politica per negare l'autorizzazione a una KMS chiave senza l'autorizzazione di una politica chiave.) La politica chiave predefinita abilita IAM le politiche. Per abilitare IAM le politiche nella tua politica chiave, aggiungi la dichiarazione politica descritta inConsente l'accesso Account AWS e abilita le politiche IAM.

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

Argomenti

- Creazione di una policy delle chiavi
- Policy delle chiavi predefinita
- Visualizza una politica chiave
- Modificare una politica chiave
- Autorizzazioni per i AWS servizi nelle politiche chiave

Creazione di una policy delle chiavi

È possibile creare e gestire le policy chiave nella AWS KMS console o utilizzando AWS KMS API operazioni come CreateKeyReplicateKey, e PutKeyPolicy.

Quando si crea una KMS chiave nella AWS KMS console, la console illustra i passaggi per creare una politica chiave basata sulla politica chiave predefinita per la console. Quando si utilizza `CreateKey` o `ReplicateKey`APIs, se non si specifica una politica chiave, viene APIs applicata la politica chiave predefinita per le chiavi create a livello di codice. Quando si utilizza il `PutKeyPolicy`API, è necessario specificare una politica 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Describe the policy statement",
      "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 proibiti nell'`Sid`elemento di un documento di IAM policy.)

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

- Caratteri stampabili ASCII
- 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 in un `Sid` elemento di IAM policy.)

Effetto

(Obbligatorio) Specifica se concedere o negare le autorizzazioni nell'istruzione di policy. I valori validi sono `Allow` e `Deny`. Se non si consente esplicitamente l'accesso a una KMS chiave, l'accesso viene negato implicitamente. Puoi anche 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. È possibile specificare IAM utenti Account AWS, IAM ruoli e alcuni AWS servizi come principali in una politica chiave. IAMi gruppi di utenti non sono un principio valido in nessun tipo di politica.

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 KMS chiave, a meno che un'altra informativa non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua KMS chiave ogni volta che dispongono delle autorizzazioni corrispondenti nel proprio account.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

Quando il principale in una dichiarazione politica chiave è un Account AWS principio espresso come`arn:aws:iam::111122223333:root"`, la dichiarazione politica non concede l'autorizzazione a nessun IAM preside. Al contrario, dà il Account AWS permesso IAM di utilizzare le politiche 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 e la politica KMS 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 KMS chiave, devi fornire l'autorizzazione in una politica chiave e in una IAM politica dell'altro account. Per informazioni dettagliate, consultare Consentire agli utenti di altri account di utilizzare una KMS chiave.

Azione

(Obbligatorio) Specificate le API operazioni da consentire o negare. Ad esempio, l'`kms:Encrypt`azione 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 KMS chiave.
> Quando manca un `Action` elemento in una dichiarazione politica chiave, la AWS KMS console segnala correttamente un errore, ma la risposta viene CreateKeycompletata PutKeyPolicyAPIscorrettamente, anche se la dichiarazione politica è inefficace.

Risorsa

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

> **Note**
>
> Se l'elemento `Resource` obbligatorio non è presente nell'istruzione della policy delle chiavi, l'istruzione non avrà alcun effetto. Una dichiarazione politica chiave senza un `Resource` elemento non si applica a nessuna KMS chiave.
> Quando manca un `Resource` elemento in una dichiarazione politica chiave, la AWS KMS console segnala correttamente un errore, ma la risposta viene

CreateKeycompletata PutKeyPolicyAPIscorrettamente, anche se la dichiarazione politica è inefficace.

Condition

(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 API richiesta 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 KMS chiave basata su tag e alias. Per informazioni dettagliate, consultare ABACper 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 AWS policy, consulta AWS IAMPolicy Reference nella Guida per l'utente. IAM

Esempi di policy delle chiavi

L'esempio seguente mostra una politica di chiave completa per una chiave di crittografia simmetrica. KMS 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 Consente all'account e ai relativi amministratori, incluso l'utente root dell'account (in caso di emergenza), di utilizzare IAM le politiche dell'account per consentire l'accesso alla chiave. KMS

- Consente al `ExampleAdminRole` IAM ruolo di amministrare la chiave. KMS
- Consente al `ExampleUserRole` IAM ruolo di utilizzare la KMS chiave.

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

```
            "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": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "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 KMS chiave, è possibile specificare la politica chiave per la nuova KMS chiave. Se non ne fornisci una, ne AWS KMS crea una per te. La politica delle chiavi predefinita AWS KMS utilizzata varia a seconda che si crei la chiave nella AWS KMS console o si utilizzi la AWS KMS API.

Politica chiave predefinita quando si crea una KMS chiave a livello di codice

Quando si crea una KMS chiave a livello di codice con AWS KMS API(anche utilizzando AWS Command Line Interfaceo AWS Tools for PowerShell) e non si specifica una politica chiave, AWS KMS applica una politica di chiave predefinita molto semplice. AWS SDKs Questa politica di chiave predefinita ha un'unica dichiarazione che fornisce al Account AWS proprietario l'autorizzazione a utilizzare IAM le politiche per consentire l'accesso a tutte le AWS KMS operazioni sulla KMS chiave. KMS Per ulteriori informazioni su questa istruzione di policy, consulta Consente l'accesso Account AWS e abilita le politiche IAM.

Politica chiave predefinita quando si crea una KMS chiave con AWS Management Console

Quando si crea una KMS chiave con AWS Management Console, la politica chiave inizia con la dichiarazione di politica che consente l'accesso Account AWS e abilita IAM le politiche. La console aggiunge quindi un'istruzione Key Administrators, un'istruzione key users e (per la maggior parte dei tipi di chiavi) un'istruzione che consente ai responsabili di utilizzare la KMS chiave con altri AWS servizi. È possibile utilizzare le funzionalità della AWS KMS console per specificare gli IAM utenti e Account AWS chi sono gli amministratori chiave e gli utenti chiave (o entrambi). IAMroles

Autorizzazioni

- Consente l'accesso Account AWS e abilita le politiche IAM
- Consente agli amministratori chiave di amministrare la chiave KMS
- Consente agli utenti chiave di utilizzare la chiave KMS
  - Consente agli utenti chiave di utilizzare una KMS chiave per operazioni crittografiche
  - Consente agli utenti chiave di utilizzare la KMS chiave con AWS i servizi

Consente l'accesso Account AWS e abilita le politiche IAM

La seguente istruzione delle policy delle chiavi predefinita è fondamentale.

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

  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 IAM di utilizzare le politiche per consentire l'accesso alla KMS chiave, oltre alla politica chiave.

  Senza questa autorizzazione, IAM le politiche che consentono l'accesso alla chiave sono inefficaci, sebbene IAM le politiche che negano l'accesso alla chiave siano comunque efficaci.
- 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 dichiarazione chiave è l'intera politica chiave predefinita per le KMS chiavi create a livello di codice. È la prima dichiarazione di policy nella policy chiave predefinita per KMS le chiavi create nella AWS KMS console.

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

Consente ai IAM criteri di consentire l'accesso alla KMS chiave.

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

Il principale in questa dichiarazione politica chiave è l'account principal, che è rappresentato da un ARN in questo formato:`arn:aws:iam::account-id:root`. L'intestatario dell'account rappresenta l' AWS account e i suoi amministratori.

Quando l'intestatario di un'informativa chiave è l'intestatario dell'account, l'informativa non fornisce alcuna autorizzazione IAM principale all'utilizzo della KMS chiave. Al contrario, consente all'account IAM di utilizzare le politiche per delegare le autorizzazioni specificate nell'informativa. Questa dichiarazione chiave predefinita consente all'account di utilizzare IAM le politiche per delegare le autorizzazioni per tutte le azioni (`kms:*`) sulla chiave. KMS

Riduce il rischio che la KMS chiave 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 necessario fornire al titolare dell'account, o a qualsiasi altro titolare, l'accesso alla KMS chiave. Tuttavia, l'accesso al principale dell'account aiuta a evitare che la chiave diventi ingestibile.

Ad esempio, supponiamo di creare una politica di chiave che consenta a un solo utente di accedere alla KMS chiave. Se poi elimini quell'utente, la chiave diventa ingestibile e devi contattare l'AWS assistenza per riottenere l'accesso alla chiave. KMS

L'informativa chiave riportata sopra consente di controllare la chiave dell'account principale, che rappresenta l'accountAccount AWS e i relativi 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 il Account AWS. IAMle migliori pratiche scoraggiano l'agire 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 KMS chiave.

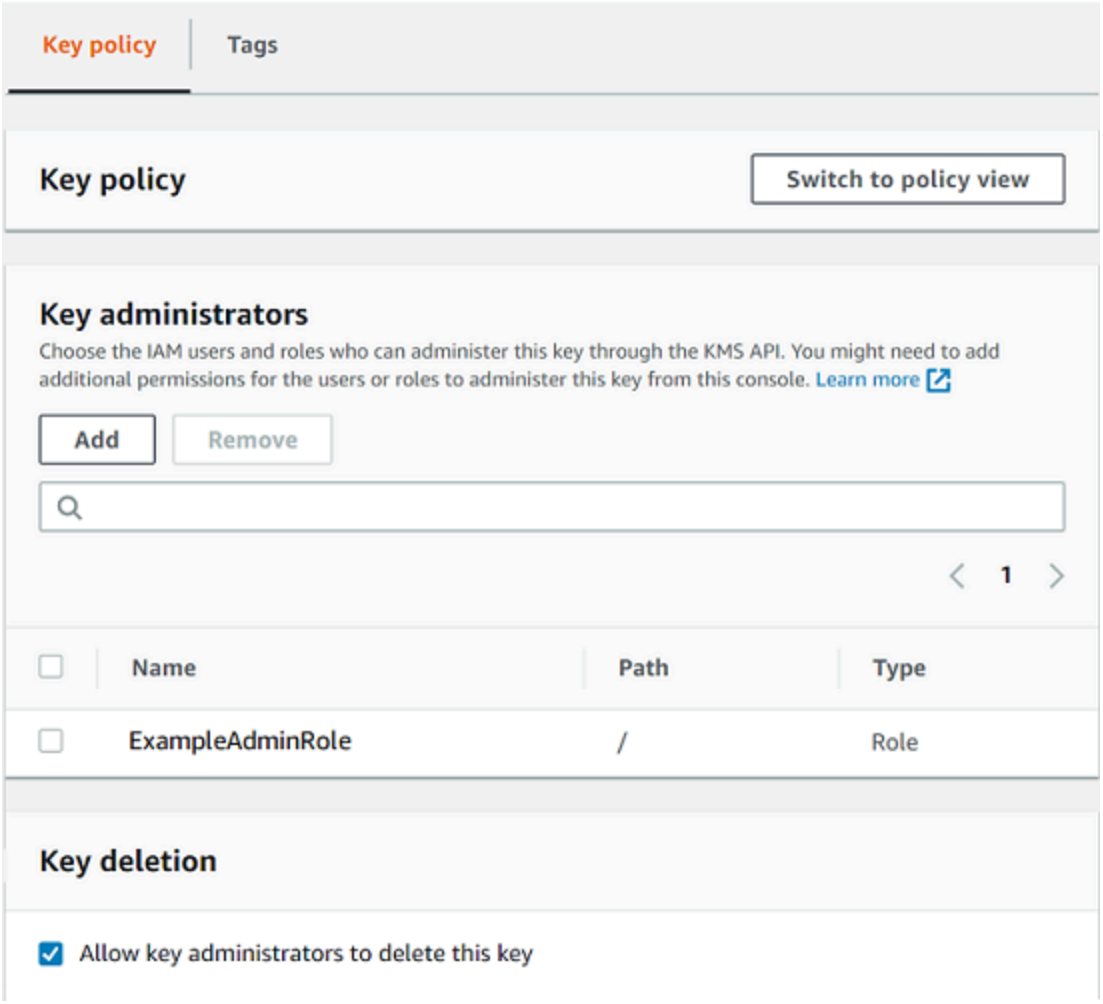
Consente agli amministratori chiave di amministrare la chiave KMS

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 KMS chiave, ma non dispongono delle autorizzazioni per utilizzare la KMS chiave nelle operazioni crittografiche. È possibile aggiungere IAM utenti e ruoli all'elenco degli amministratori chiave quando si crea la KMS chiave nella visualizzazione predefinita o nella visualizzazione delle politiche.

> **Warning**
>
> Poiché gli amministratori chiave sono autorizzati a modificare la politica chiave e creare sovvenzioni, possono concedere a se stessi e ad altri AWS KMS autorizzazioni non specificate in questa politica.
> I responsabili che dispongono del permesso di gestire tag e alias possono anche controllare l'accesso a una chiave. KMS Per informazioni dettagliate, consultare ABACper AWS KMS.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

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



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

> **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 Key Administrators predefinita per la chiave più comune, una KMS chiave di crittografia KMS simmetrica a regione singola, consente le seguenti autorizzazioni. Per informazioni dettagliate su ciascuna autorizzazione, consultare AWS KMS autorizzazioni.

Quando si utilizza la AWS KMS console per creare una KMS chiave, la console aggiunge gli utenti e i ruoli specificati all'`Principal`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 API operazioni,

gli amministratori chiave possono utilizzarle automaticamente. Non è necessario aggiornare le policy della chiave per includere le nuove operazioni. Se preferisci limitare i tuoi amministratori chiave a un insieme fisso di API operazioni, puoi modificare la tua politica chiave.

`kms:Create*`

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

`kms:Describe*`

Consente `kms:DescribeKey`. L'`kms:DescribeKey`autorizzazione è richiesta per visualizzare la pagina dei dettagli chiave di una KMS chiave in AWS Management Console.

`kms:Enable*`

Consente `kms:EnableKey`. Per le KMS chiavi di crittografia simmetriche, consente anche. `kms:EnableKeyRotation`

`kms:List*`

Permette `kms:ListGrants`, `kms:ListKeyPolicies` e `kms:ListResourceTags`. (Le `kms:ListKeys` autorizzazioni `kms:ListAliases` and, necessarie per visualizzare KMS le chiavi in AWS Management Console, sono valide solo nelle IAM politiche.)

`kms:Put*`

Consente `kms:PutKeyPolicy`. Questa autorizzazione consente agli amministratori chiave di modificare la politica chiave per questa KMS chiave.

`kms:Update*`

Consente `kms:UpdateAlias` e `kms:UpdateKeyDescription`. Per le chiavi multiregionali, consente l'utilizzo di `kms:UpdatePrimaryRegion`questa KMS chiave.

`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 KMS chiavi di crittografia simmetriche, consente anche. `kms:DisableKeyRotation`

`kms:Get*`

Consente `kms:GetKeyPolicy` e `kms:GetKeyRotationStatus`. Per KMS le chiavi con materiale chiave importato, consente. `kms:GetParametersForImport` Per le KMS chiavi asimmetriche, lo consente. `kms:GetPublicKey` L'`kms:GetKeyPolicy`autorizzazione è necessaria per visualizzare la politica chiave di una KMS chiave in. AWS Management Console

`kms:Delete*`

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

`kms:TagResource`

Consente `kms:TagResource`, che consente agli amministratori chiave di aggiungere tag alla KMS chiave. Poiché i tag possono essere utilizzati anche per controllare l'accesso alla KMS chiave, questa autorizzazione può consentire agli amministratori di consentire o negare l'accesso alla chiave. KMS Per informazioni dettagliate, consultare ABACper AWS KMS.

`kms:UntagResource`

Consente `kms:UntagResource`, che consente agli amministratori chiave di eliminare i tag dalla chiave. KMS Poiché i tag possono essere utilizzati per controllare l'accesso alla chiave, questa autorizzazione può consentire agli amministratori di consentire o negare l'accesso alla chiave. KMS Per informazioni dettagliate, consultare ABACper AWS KMS.

`kms:ScheduleKeyDeletion`

Consente `kms:ScheduleKeyDeletion`, che consente agli amministratori chiave di eliminare questa chiave. KMS Per eliminare questa autorizzazione, deseleziona l'opzione Consenti agli amministratori della chiave di eliminare questa chiave.

`kms:CancelKeyDeletion`

Consente `kms:CancelKeyDeletion`, che consente agli amministratori chiave di annullare l'eliminazione di questa KMS chiave. Per eliminare questa autorizzazione, deseleziona 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 contenuto 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'`kms:ImportKeyMaterial`autorizzazione consente agli amministratori chiave di importare materiale chiave nella chiave. KMS Questa autorizzazione è inclusa nella politica chiave solo quando si crea una KMS chiave senza materiale 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 chiave di utilizzare la chiave KMS

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

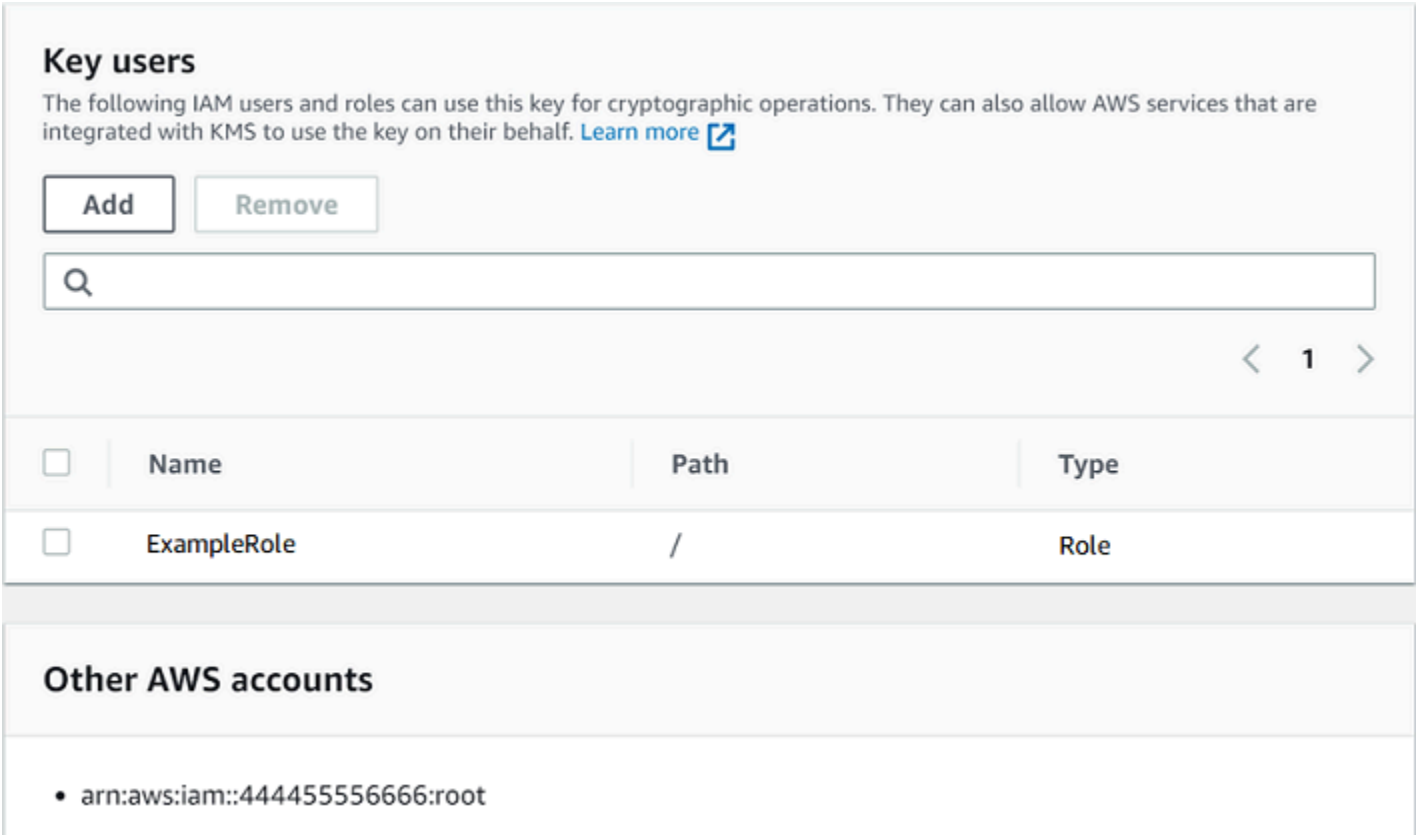
La console aggiunge due istruzioni di policy alla policy delle chiavi per gli utenti della chiave.

- Usa la KMS chiave direttamente: la prima dichiarazione sulla politica chiave consente agli utenti chiave di utilizzare direttamente la KMS chiave per tutte le operazioni crittografiche supportate per quel tipo di KMS chiave.
- Usa la KMS chiave con AWS i servizi: la seconda dichiarazione politica concede agli utenti chiave il permesso di consentire ai AWS servizi integrati di utilizzare la KMS chiave AWS KMS per loro conto per proteggere le risorse, come i bucket Amazon S3 e le tabelle Amazon DynamoDB.

Puoi aggiungere IAM utenti, IAM ruoli e altro 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 utenti Account AWS di utilizzare la KMS chiave, consultaConsentire agli utenti di altri account di utilizzare una KMS chiave.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.



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 si utilizza la AWS KMS console per creare una KMS chiave, la console aggiunge gli utenti e i ruoli specificati all'`Principal`elemento in ogni istruzione key users.

> **Note**
>
> La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori di dichiarazione `"Allow use of the key"` 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 chiave di utilizzare una KMS chiave per operazioni crittografiche

Gli utenti chiave sono autorizzati a utilizzare la KMS chiave direttamente in tutte le operazioni crittografiche supportate sulla KMS chiave. Possono inoltre utilizzare l'DescribeKeyoperazione per ottenere informazioni dettagliate sulla KMS chiave nella AWS KMS console o utilizzando le AWS KMS API operazioni.

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

Chiavi di crittografia simmetriche KMS

La console aggiunge la seguente dichiarazione alla politica chiave per le chiavi di crittografia simmetriche. KMS

```
{
  "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": "*"
}
```

HMACKMSchiavi

La console aggiunge la seguente dichiarazione alla politica chiave per HMAC KMS le chiavi.

```
{
  "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 asimmetriche per la crittografia a KMS chiave pubblica

La console aggiunge la seguente dichiarazione alla politica chiave per le chiavi asimmetriche con un utilizzo chiave di KMS Encrypt and decrypt.

```
{
  "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 asimmetriche per la firma e la verifica KMS

La console aggiunge la seguente dichiarazione alla politica chiave per le chiavi asimmetriche KMS con un utilizzo chiave di Sign and verify.

```
{
  "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 asimmetriche KMS per derivare segreti condivisi

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

Consente agli utenti chiave di crittografare i dati con questa chiave. KMS

`kms:Decrypt`

Consente agli utenti chiave di decrittografare i dati con questa chiave. KMS

`kms:DeriveSharedSecret`

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

`kms:DescribeKey`

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

`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 API operazioni: GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair, e GenerateDataKeyPairWithoutPlaintext. Queste autorizzazioni sono valide solo sulle chiavi simmetriche che crittografano KMS le chiavi dati.

km: GenerateMac

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

km: GetPublicKey

Consente agli utenti chiave di scaricare la chiave pubblica della chiave asimmetricaKMS. Le parti con cui condividi questa chiave pubblica possono crittografare i dati all'esterno di. AWS KMS Questi testi cifrati possono essere decriptati solo chiamando l'operazione Decrypt in AWS KMS.

km: * ReEncrypt

Consente agli utenti chiave di crittografare nuovamente i dati originariamente crittografati con questa KMS chiave o di utilizzare questa KMS chiave per ricrittografare dati precedentemente crittografati. L'ReEncryptoperazione richiede l'accesso sia alle chiavi di origine che a quelle di destinazione. KMS A tale scopo, è possibile concedere l'`kms:ReEncryptFrom`autorizzazione sulla KMS chiave di origine e l'`kms:ReEncryptTo`autorizzazione sulla KMS chiave di destinazione. Tuttavia, per semplicità, la console consente `kms:ReEncrypt*` (con il carattere * jolly) di utilizzare entrambi i KMS tasti.

kms:Sign

Consente agli utenti chiave di firmare i messaggi con questa KMS chiave.

kms:Verify

Consente agli utenti chiave di verificare le firme con questa KMS chiave.

km: VerifyMac

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

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

La politica delle chiavi predefinita nella console offre inoltre agli utenti chiave le autorizzazioni necessarie 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 KMS chiave, ma solo quando la richiesta di operazione di concessione proviene da un AWS 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 KMS chiave per proteggere i dati dell'utente.

Gli utenti chiave richiedono queste autorizzazioni di concessione per utilizzare la propria KMS chiave con servizi integrati, ma tali 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 chiave possono utilizzare queste autorizzazioni sulla KMS chiave nei seguenti modi.

- Usa questa KMS chiave con Amazon Elastic Block Store (AmazonEBS) e Amazon Elastic Compute Cloud (AmazonEC2) per collegare un EBS volume crittografato a un'EC2istanza. L'utente della chiave EC2 autorizza implicitamente Amazon a utilizzare la KMS chiave per collegare il volume crittografato all'istanza. Per ulteriori informazioni, consulta Come utilizza Amazon Elastic Block Store (AmazonEBS) AWS KMS.
- Usa questa KMS chiave con Amazon Redshift per avviare un cluster crittografato. L'utente chiave autorizza implicitamente Amazon Redshift a utilizzare KMS la chiave per avviare il cluster crittografato e creare istantanee crittografate. Per ulteriori informazioni, consulta Come utilizza Amazon Redshift AWS KMS.

- Utilizza questa KMS chiave con altri AWS servizi integrati con AWS KMS tali concessioni d'uso per creare, gestire o utilizzare risorse crittografate con tali 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 AWSnel tuo account utilizzando la AWS KMS console o l'GetKeyPolicyoperazione in AWS KMS API. Non è possibile utilizzare queste tecniche per visualizzare la politica chiave di una KMS chiave in un altro Account AWS.

Per ulteriori informazioni sulle politiche AWS KMS chiave, consultaPolitiche chiave in AWS KMS. Per informazioni su come determinare quali utenti e ruoli hanno accesso a una KMS chiave, consultathe section called “Determinazione dell'accesso”.

Utilizzo della AWS KMS console

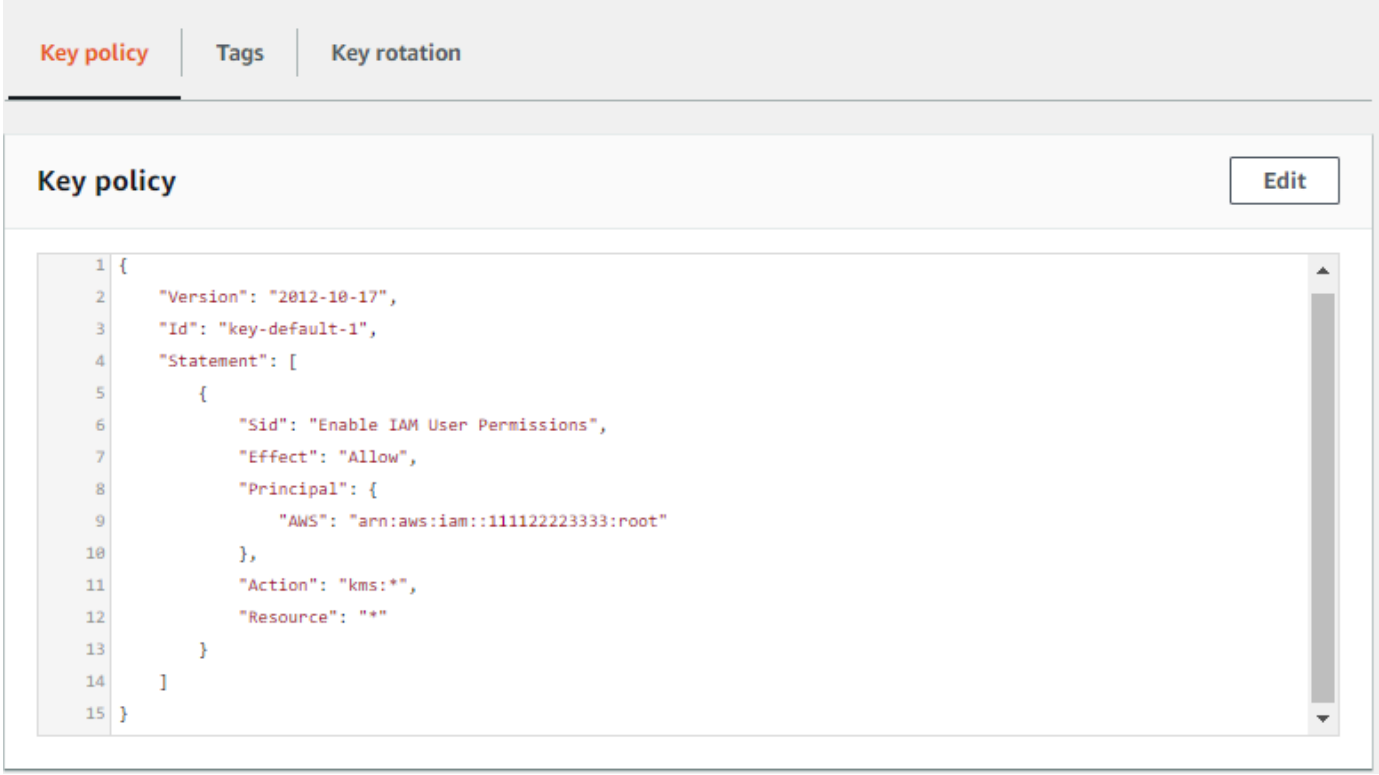
Gli utenti autorizzati possono visualizzare la policy delle chiavi per unaChiave gestita da AWS o una chiave gestita dal cliente nella scheda Key policy (Policy delle chiavi) della AWS Management Console.

Per visualizzare la politica chiave per una KMS chiave in AWS Management Console, devi disporre delle autorizzazioni kms: ListAliases, kms: DescribeKey e kms:. GetKeyPolicy

1. Accedi a AWS Management Console 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 delle KMS chiavi, scegli l'alias o l'ID della KMS chiave che desideri esaminare.
5. Scegli la scheda Policy della chiave.

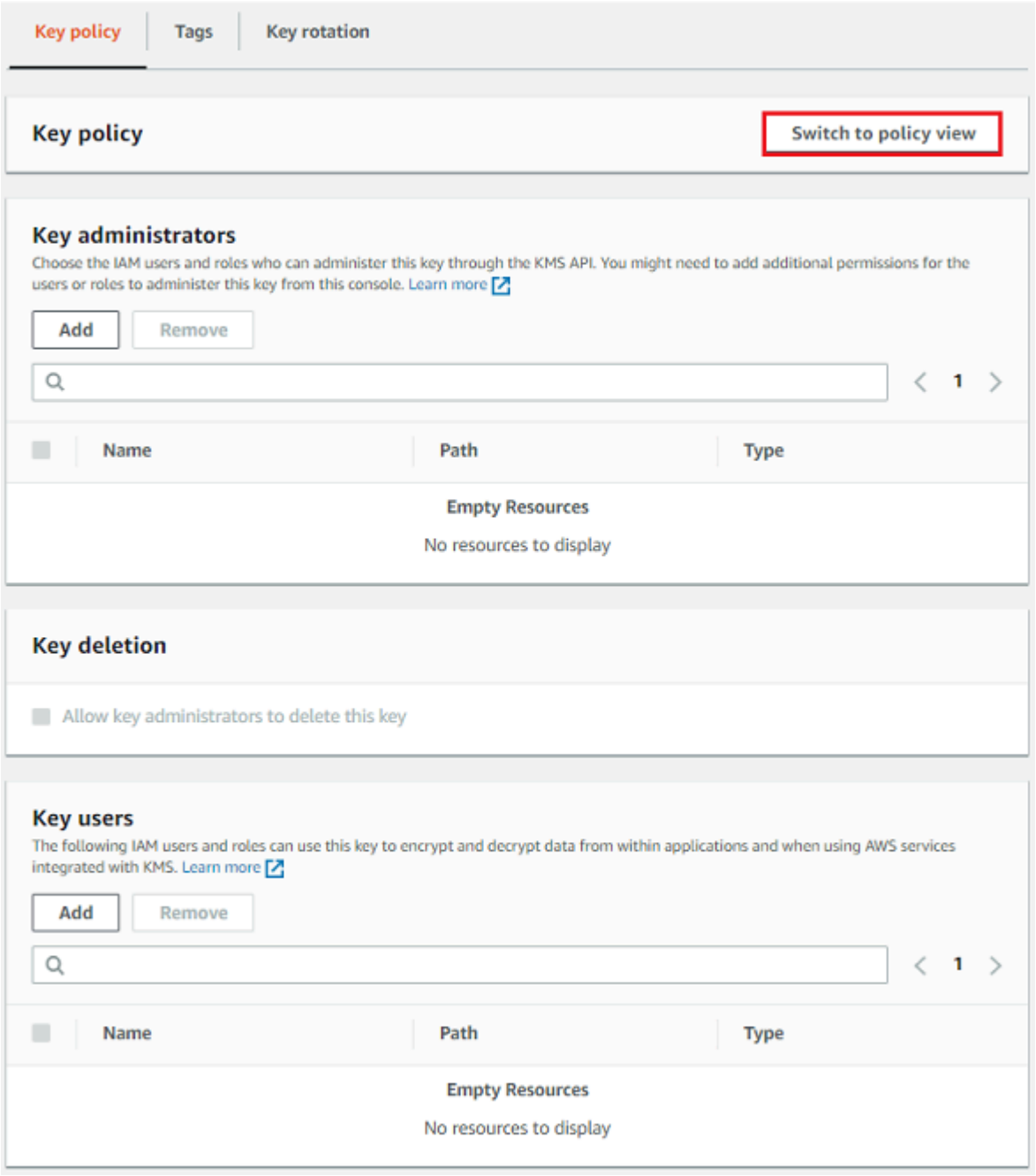
Nella tab Policy chiave è possibile che venga visualizzato il documento della policy delle chiavi. Si tratta della visualizzazione policy. Nelle dichiarazioni politiche chiave, puoi vedere i responsabili a cui è stato concesso l'accesso alla KMS chiave dalla politica chiave e puoi vedere le azioni che possono eseguire.

Nell'esempio seguente viene illustrata la visualizzazione policy per la policy di chiave predefinita.



In alternativa, se hai creato la KMS chiave in AWS Management Console, 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.



Utilizzando il AWS KMS API

Per ottenere la politica chiave per una KMS chiave nel tuo Account AWS, usa l'GetKeyPolicyoperazione in 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-policycomando contenuto in AWS Command Line Interface (AWS CLI), ma è possibile utilizzare qualsiasi comando 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 in formato testoJSON, anziché per facilitarne la visualizzazione.

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.

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

Modificare una politica chiave

È possibile modificare la politica chiave per una KMS chiave del Account AWS sistema utilizzando l'PutKeyPolicyoperazione AWS Management Console o. Non è possibile utilizzare queste tecniche per modificare la politica chiave di una KMS chiave in un'altra Account AWS.

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 KMS chiave nell'account. Non puoi visualizzare o modificare la policy delle chiavi per una Chiave di proprietà di AWS.
- Puoi aggiungere o rimuovere IAM utenti, IAM ruoli e Account AWS nella politica 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 è possibile aggiungere IAM gruppi a una politica chiave, ma è possibile aggiungere più IAM utenti e IAM ruoli. Per ulteriori informazioni, consulta Consentire a più IAM presidi di accedere a una chiave KMS.
- Se si aggiungono elementi esterni Account AWS a una politica chiave, è necessario utilizzare anche i IAM criteri negli account esterni per concedere autorizzazioni a IAM utenti, gruppi o ruoli in tali account. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.
- 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 chiavi in tre diversi modi, ognuno dei quali è illustrato nelle seguenti sezioni.

Argomenti

- Utilizzo della visualizzazione predefinita della AWS Management Console
- Utilizzo della visualizzazione delle politiche AWS Management Console
- Utilizzando il AWS KMS API

Utilizzo della visualizzazione predefinita della AWS Management Console

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 AWS Management Console o Utilizzando il AWS KMS API.

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 chiave e consentire o impedire agli amministratori chiave di eliminare la KMS chiave, utilizza i controlli nella sezione Amministratori chiave

della pagina. Gli amministratori chiave gestiscono la KMS chiave, inclusa l'attivazione e la disabilitazione, l'impostazione dei criteri chiave e l'abilitazione della rotazione delle chiavi.

- Per aggiungere o rimuovere utenti chiave e consentire o impedire l'uso della KMS chiave Account AWS da parte di utenti esterni, utilizza i controlli nella sezione Utenti chiave della pagina. Gli utenti chiave possono utilizzare la KMS chiave in operazioni crittografiche, come la crittografia, la decrittografia, la ricrittografia e la generazione di chiavi di dati.

Utilizzo della visualizzazione delle politiche AWS Management Console

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. Rivedi 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, selezionate 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).

Utilizzando il AWS KMS API

È possibile utilizzare l'PutKeyPolicyoperazione per modificare la politica chiave di una KMS chiave nel proprio Account AWS. Non è possibile utilizzarla API su una KMS chiave diversa Account AWS.

1. Utilizzate l'GetKeyPolicyoperazione per ottenere il documento di policy chiave esistente, quindi salvate il documento di policy chiave in un file. Per il codice di esempio in più linguaggi di programmazione, consulta Utilizzare `GetKeyPolicy` con un AWS SDK o CLI.
2. Aprire il documento di policy delle chiavi in un editor di testo, modificarlo e salvare il file.
3. Utilizzare l'PutKeyPolicyoperazione per applicare alla chiave il documento di policy KMS chiave aggiornato. Per il codice di esempio in più linguaggi di programmazione, consulta Utilizzare `PutKeyPolicy` con un AWS SDK o CLI.

Per un esempio di copia di una politica chiave da una KMS chiave all'altra, vedete l'GetKeyPolicy esempio nel AWS CLI Command Reference.

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 AWSo Chiavi gestite da AWS, il servizio stabilisce e mantiene le politiche chiave per queste KMS chiavi.

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. Ad esempio, per le autorizzazioni richieste da Amazon Elastic Block Store (AmazonEBS), consulta Autorizzazioni per IAM gli utenti nella Amazon User Guide e Amazon EC2 User Guide. EC2 Per le autorizzazioni richieste da Secrets Manager, vedere Autorizzazione all'uso della KMS chiave nella Guida per l'AWS Secrets Manager utente.

Utilizzo IAM delle politiche con AWS KMS

Puoi utilizzare IAM le policy, insieme alle policy chiave, alle concessioni e alle policy degli VPC endpoint, per controllare l'accesso al tuo account. AWS KMS keys AWS KMS

> **Note**
>
> Per utilizzare una IAM policy per controllare l'accesso a una KMS chiave, la policy chiave relativa alla KMS chiave deve autorizzare l'account a utilizzare IAM le policy. In particolare, la policy della chiave deve includere l' istruzione di policy che abilita le policy IAM.
> Questa sezione spiega come utilizzare IAM le policy per controllare l'accesso alle AWS KMS operazioni. Per informazioni più generali suIAM, consulta la Guida IAM per l'utente.

Tutte le KMS chiavi devono avere una politica chiave. IAMle politiche sono facoltative. Per utilizzare una IAM politica per controllare l'accesso a una KMS chiave, la politica chiave relativa alla KMS chiave deve consentire all'account l'autorizzazione a utilizzare IAM le politiche. In particolare, la policy della chiave deve includere l' istruzione di policy che abilita le policy IAM.

IAMle politiche possono controllare l'accesso a qualsiasi AWS KMS operazione. A differenza delle policy chiave, IAM le policy possono controllare l'accesso a più KMS chiavi e fornire le autorizzazioni per il funzionamento di diversi AWS servizi correlati. Tuttavia, IAM le policy sono particolarmente utili per controllare l'accesso a operazioni CreateKey, ad esempio quelle operazioni che non possono essere controllate da una policy chiave perché non coinvolgono alcuna KMS chiave particolare.

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

Per assistenza nella scrittura e nella formattazione di un documento di JSON policy, consulta il IAMJSONPolicy Reference nella Guida per l'IAMutente.

Puoi utilizzare policy IAM nei seguenti modi:

- Associare una politica di autorizzazioni a un ruolo per le autorizzazioni federative o interaccount: puoi allegare una IAM politica a un IAM ruolo 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 dei IAM ruoli, consulta Ruoli nella Guida per l'utente. IAM 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 IAM

best practice consigliano di utilizzare identità con credenziali temporanee, come IAM i ruoli, quando possibile.

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

```
{
  "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 si allega una IAM politica a un'IAMidentità, tale identità ottiene le autorizzazioni specificate nella politica.

Per una tabella che mostra tutte le AWS KMS API azioni e le risorse a cui si applicano, consulta laRiferimento per le autorizzazioni.

Consentire a più IAM presidi di accedere a una chiave KMS

I gruppi IAM non sono principali validi in una policy delle chiavi. Per consentire a più utenti e ruoli di accedere a una KMS chiave, esegui una delle seguenti operazioni:

- Utilizzate un IAM ruolo come principale nella politica chiave. Più utenti autorizzati possono assumere il ruolo secondo necessità. Per i dettagli, consulta IAMi ruoli nella Guida IAM per l'utente.

  Sebbene sia possibile elencare più IAM utenti in una politica chiave, questa pratica non è consigliata perché richiede l'aggiornamento della politica chiave ogni volta che l'elenco degli utenti autorizzati cambia. Inoltre, le IAM best practice scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
- Utilizza una IAM politica per concedere l'autorizzazione a un IAM gruppo. A tale scopo, assicurati che la politica chiave includa l'istruzione che abilita IAM le politiche per consentire l'accesso alla

KMS chiave, crea una IAM politica che consenta l'accesso alla KMS chiave e quindi associa tale politica a un IAM gruppo che contiene gli IAM utenti 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 i gruppi di IAM utenti nella Guida IAM per l'utente

Per ulteriori informazioni su come le politiche e le IAM politiche AWS KMS chiave interagiscono, vedereRisoluzione dei problemi relativi alle AWS KMS autorizzazioni.

Best practice per le policy IAM

Garantire l'accesso a AWS KMS keys è fondamentale per la sicurezza di tutte le AWS risorse. KMSle chiavi vengono utilizzate per proteggere molte delle risorse più sensibili del tuo Account AWS. Prenditi il tempo necessario per progettare le politiche chiave, IAM le politiche, le concessioni e le politiche degli VPC endpoint che controllano l'accesso alle tue KMS chiavi.

Nelle dichiarazioni IAM politiche che controllano l'accesso alle KMS chiavi, utilizza il principio dei meno privilegiati. IAMAssegna ai responsabili solo le autorizzazioni di cui hanno bisogno solo sulle KMS chiavi che devono usare o gestire.

Le seguenti best practice si applicano alle IAM politiche che controllano l'accesso a AWS KMS chiavi e alias. Per indicazioni generali IAM sulle best practice relative alle policy, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida per l'IAMutente.

Utilizzo delle policy delle chiavi

Quando possibile, fornisci le autorizzazioni nelle policy chiave che riguardano una KMS chiave, anziché in una IAM policy che può essere applicata a molte KMS chiavi, incluse quelle di altre Account AWS. Ciò è particolarmente importante per le autorizzazioni sensibili come kms: PutKeyPolicy e kms: ScheduleKeyDeletion ma anche per le operazioni crittografiche che determinano la protezione dei dati.

CreateKey Limita le autorizzazioni

Concedi il permesso di creare chiavi (kms: CreateKey) solo ai principali che ne hanno bisogno. I responsabili che creano una KMS chiave ne stabiliscono anche la politica, in modo che possano concedere a se stessi e agli altri il permesso di usare e gestire le KMS chiavi 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 di crittografia simmetriche. KMS

Specificare le KMS chiavi in una politica IAM

Come procedura consigliata, specificare la chiave ARN di ogni KMS chiave a cui si applica l'autorizzazione nell'`Resource`elemento della dichiarazione politica. Questa pratica limita l'autorizzazione alle KMS chiavi richieste dal principale. Ad esempio, questo `Resource` elemento elenca solo le KMS chiavi che il principale deve usare.

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

Non è possibile utilizzare un ID chiave, un nome alias o un alias ARN per rappresentare una KMS chiave nel `Resource` campo di una IAM politica. Se si specifica un aliasARN, la politica si applica all'alias, non alla chiave. KMS Per informazioni sulle IAM politiche per gli alias, vedere Controllo dell'accesso agli alias

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

Utilizza i caratteri jolly (*) in modo giudizioso. In una politica chiave, il carattere jolly nell'`Resource`elemento rappresenta la KMS chiave a cui è associata la politica chiave. Tuttavia, in una IAM policy, solo un carattere jolly presente nell'`Resource`elemento (`"Resource": "*"`) applica le autorizzazioni a tutte le KMS chiavi di tutte le chiavi Account AWS che l'account del principale è autorizzato a utilizzare. Ciò potrebbe includere KMSle chiavi di un altro Account AWS account, oltre alle KMS chiavi dell'account del principale.

Ad esempio, per utilizzare una KMS chiave in un'altra Account AWS, un principale necessita dell'autorizzazione della politica KMS chiave della chiave dell'account esterno e di una IAM politica del proprio account. Supponiamo che un account arbitrario abbia dato al tuo account Account AWS KMS:Decrypt l'autorizzazione sulle proprie chiavi. KMS In tal caso, una IAM politica del vostro account che conceda un'`kms:Decrypt`autorizzazione di ruolo su tutte le KMS chiavi (`"Resource": "*"`) soddisferebbe la parte del requisito. IAM Di conseguenza, i responsabili che possono assumere quel ruolo possono ora decrittografare i testi cifrati utilizzando la KMS

chiave dell'account non attendibile. Le voci relative alle loro operazioni vengono visualizzate nei registri di entrambi gli account. CloudTrail

In particolare, evitate di utilizzare `"Resource": "*"` in una dichiarazione politica che consenta le seguenti API operazioni. Queste operazioni possono essere richiamate sui KMS tasti di altri Account AWS.

- DescribeKey
- GetKeyRotationStatus
- Operazioni crittografiche (Encrypt, Decrypt,,,,, GenerateDataKey,, GenerateDataKeyPairGenerateDataKeyWithoutPlaintext, Sign GenerateDataKeyPairWithoutPlaintextGetPublicKeyReEncrypt, Verify)
- 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:UpdateAlias, kms: DeleteAlias) devono essere allegate all'alias e alla chiave. KMS È possibile utilizzarli `"Resource": "*"` in una IAM politica per rappresentare gli alias e le chiavi o specificare gli alias e KMS le chiavi nell'elemento. KMS `Resource` Per alcuni esempi, consulta Controllo dell'accesso agli alias.

Gli esempi in questo argomento forniscono ulteriori informazioni e linee guida per la progettazione IAM di politiche per KMS le chiavi. Per le IAM best practice relative a tutte le AWS risorse, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida per IAM l'utente.

Specificazione KMS delle chiavi nelle dichiarazioni IAM politiche

È possibile utilizzare una IAM politica per consentire a un principale di utilizzare o gestire KMS le chiavi. KMSle chiavi sono specificate nell'`Resource`elemento della dichiarazione politica.

- Per specificare una KMS chiave in una dichiarazione IAM politica, è necessario utilizzare la relativa chiave ARN. Non è possibile utilizzare un ID chiave, un nome alias o un alias ARN per identificare una KMS chiave in una dichiarazione di IAM policy.

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

  Per controllare l'accesso a una KMS chiave in base ai relativi alias, usa i tasti kms: RequestAlias o kms: condition. ResourceAliases Per informazioni dettagliate, consultare ABACper AWS KMS.

  Usa un ARN alias come risorsa solo in un'informativa che controlla l'accesso alle operazioni di alias, ad esempio, o. CreateAliasUpdateAliasDeleteAlias Per informazioni dettagliate, consultare Controllo dell'accesso agli alias.
- Per specificare più KMS chiavi nell'account e nella regione, utilizza caratteri jolly (*) nelle posizioni dell'area o dell'ID della risorsa della chiave. ARN

  Ad esempio, per specificare tutte le KMS chiavi nella regione Stati Uniti occidentali (Oregon) di un account, utilizza "»`Resource": "arn:aws:kms:us-west-2:111122223333:key/*`. Per specificare tutte le KMS chiavi in tutte le regioni dell'account, usa `"Resource": "arn:aws:kms:*:111122223333:key/*`».
- Per rappresentare tutte le KMS chiavi, usa solo un carattere jolly (`"*"`). Utilizzate questo formato per operazioni che non utilizzano alcuna KMS chiave particolare CreateKey, GenerateRandomListAliases, e ListKeys.

Quando si scrivono le dichiarazioni politiche, è consigliabile specificare solo KMS le chiavi che il principale deve utilizzare, anziché concedere loro l'accesso a tutte le KMS chiavi.

Ad esempio, la seguente dichiarazione IAM politica consente al principale di richiamare le operazioni DescribeKeyGenerateDataKey, Decrypt solo sulle KMS chiavi elencate nell'`Resource`elemento

dell'informativa. La specificazione KMS delle chiavi per chiaveARN, che è una procedura consigliata, garantisce che le autorizzazioni siano limitate solo alle chiavi specificate. KMS

```
{
  "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/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

Per applicare l'autorizzazione a tutte le KMS chiavi di un particolare trust Account AWS, puoi utilizzare caratteri jolly (*) nelle posizioni Regione e ID chiave. Ad esempio, la seguente dichiarazione politica consente al principale di richiamare le operazioni specificate su tutte le KMS chiavi in due account di esempio attendibili.

```
{
  "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 KMS chiavi che l'account è autorizzato a utilizzare, è consigliata principalmente

per le operazioni senza una KMS chiave particolare e per `Deny` le istruzioni. 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 KMS chiave particolare, cerca il valore della KMSchiave nella colonna Risorse della tabella inthe section called “Riferimento per le autorizzazioni”.

Ad esempio, la seguente dichiarazione politica utilizza un `Deny` effetto per vietare ai principali di utilizzare le operazioni specificate su qualsiasi KMS chiave. Utilizza un carattere jolly nell'`Resource`elemento per rappresentare tutte le chiavi. KMS

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

La seguente dichiarazione di policy utilizza solo un carattere jolly per rappresentare tutte le KMS chiavi. Permette tuttavia solo operazioni di sola lettura meno sensibili e operazioni che non si applicano a nessuna chiave particolare. KMS

```
{
  "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 operazioni AWS KMS .

> **Important**
>
> Alcune delle autorizzazioni contenute nelle seguenti politiche sono consentite solo quando anche la politica KMS chiave della chiave le consente. Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Per informazioni sulla stesura e la formattazione di un documento di JSON policy, consulta il riferimento alle IAM JSON policy nella Guida per l'IAMutente.

Esempi

- Consenti a un utente di visualizzare KMS le chiavi nella console AWS KMS
- Consenti a un utente di creare chiavi KMS
- Consenti a un utente di crittografare e decrittografare con qualsiasi KMS chiave in una specifica chiave Account AWS
- Consenti a un utente di crittografare e decrittografare con qualsiasi chiave in una regione e in una regione specifiche KMS Account AWS
- Consenti a un utente di crittografare e decrittografare con chiavi specifiche KMS
- Impedire a un utente di disabilitare o eliminare qualsiasi chiave KMS

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

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

Per visualizzare KMS le chiavi nelle pagine Chiavi gestite da AWSe nelle pagine delle chiavi gestite dal cliente, i principali richiedono le GetResources autorizzazioni kms: ListKeys ListAliases, kms: e tag:, 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 KMS chiave nelle pagine di dettaglio chiave. KMS Le ListRoles autorizzazioni iam: ListUsers e iam: 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 KMS chiavi negli archivi di chiavi personalizzati, i principali necessitano anche di kms: permission. DescribeCustomKeyStores

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

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

```
{
  "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": "*"
```

```
    }
  ]
}
```

Consenti a un utente di creare chiavi KMS

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

Per limitare l'utente a particolari tipi di KMS chiavi, usa le chiavi di condizione kms: KeySpec, kms: KeyUsage e kms:. KeyOrigin

```
{
  "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 iniziale della chiave per la chiave. KMS Tuttavia, il `CreateKey` chiamante deve disporre di kms: PutKeyPolicy permission, che gli consente di modificare la politica KMS chiave, oppure deve specificare il `BypassPolicyLockoutSafetyCheck` parametro di`CreateKey`, che non è consigliato. Il `CreateKey` chiamante può ottenere l'`kms:PutKeyPolicy`autorizzazione per la KMS chiave da una IAM policy oppure può includere questa autorizzazione nella policy chiave della KMS chiave che sta creando.
- kms: TagResource — Per aggiungere tag alla KMS chiave durante l'`CreateKey`operazione, il `CreateKey` chiamante deve avere l'TagResourceautorizzazione kms: in una policy. IAM Includere questa autorizzazione nella politica chiave della nuova KMS chiave non è sufficiente. Tuttavia, se il `CreateKey` chiamante include `kms:TagResource` la politica chiave iniziale, può aggiungere tag in una chiamata separata dopo la creazione della KMS chiave.
- kms: CreateAlias — I principali che creano una KMS chiave nella AWS KMS console devono avere l'CreateAliasautorizzazione kms: sulla KMS chiave 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 KMS chiave in una politica o in una politica chiave. IAM Per informazioni dettagliate, consultare Controllo dell'accesso agli alias.

Inoltre`kms:CreateKey`, la seguente IAM politica fornisce l'`kms:TagResource`autorizzazione su tutte le KMS chiavi Account AWS e l'`kms:CreateAlias`autorizzazione su tutti gli alias dell'account. 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. È consigliabile impostare queste autorizzazioni nella politica chiave, laddove si applichino esclusivamente a una KMS chiave.

```
{
  "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 KMS chiave in una specifica chiave Account AWS

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

```
{
  "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 in una regione e in una regione specifiche KMS Account AWS

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

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/*"
    ]
  }
}
```

Consenti a un utente di crittografare e decrittografare con chiavi specifiche KMS

La seguente IAM politica consente a un utente di crittografare e decrittografare i dati con le due KMS chiavi specificate nell'elemento. `Resource` Quando si specifica una KMS chiave in una dichiarazione IAM politica, è necessario utilizzare la chiave della chiave ARN. KMS

```
{
  "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/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

Impedire a un utente di disabilitare o eliminare qualsiasi chiave KMS

La seguente IAM politica impedisce a un utente di disabilitare o eliminare qualsiasi KMS chiave, anche quando un'altra IAM politica o una politica chiave consente tali autorizzazioni. 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 AWS KMS autorizzazioni.

```
{
  "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. RCPsti aiutano a limitare centralmente l'accesso esterno alle tue AWS risorse su larga scala. RCPsintegrano le politiche di controllo dei servizi (SCPs). SCPsPuò essere utilizzato per impostare centralmente le autorizzazioni massime per IAM i ruoli e gli utenti dell'organizzazione, ma RCPs può essere utilizzato per impostare centralmente le autorizzazioni massime per AWS le risorse dell'organizzazione.

È possibile utilizzare RCPs per gestire le autorizzazioni relative alle KMS chiavi gestite dai clienti nell'organizzazione. RCPsda soli non sono sufficienti a concedere le autorizzazioni alle chiavi gestite dai clienti. Nessuna autorizzazione viene concessa da un. RCP An RCP definisce una barriera di autorizzazioni, o imposta dei limiti, alle azioni che le identità possono intraprendere sulle risorse degli account interessati. L'amministratore deve comunque allegare politiche basate sull'identità ai IAM ruoli o agli utenti 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 suRCPs, consulta le politiche di controllo delle risorse nella Guida per l'AWS Organizations utente.
- Per dettagli su come definireRCPs, inclusi esempi, consultate la RCPsintassi nella Guida per l'AWS Organizations utente.

L'esempio seguente dimostra come utilizzare an per impedire RCP 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 carattereRCP, anche se l'`Action`elemento specifica un asterisco (*) come carattere jolly.
> Per ulteriori informazioni su come `kms:RetireGrant` viene determinata l'autorizzazione a, vedere. Ritirare e revocare le concessioni

```
{
    "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 KMS chiavi 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 `atrue`, causando l'`aws:SourceOrgID`applicazione della chiave.

Per ulteriori informazioni sul problema del deputato confuso, vedere Il problema del deputato confuso nella Guida per l'IAMutente.

```
{
```

```
    "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 sovvenzione è uno strumento politico che consente ai AWS mandanti di utilizzare KMS le chiavi nelle operazioni crittografiche. Può anche consentire loro di visualizzare una KMS chiave (`DescribeKey`) e creare e gestire le sovvenzioni. Quando si autorizza l'accesso a una KMS chiave, le sovvenzioni vengono prese in considerazione insieme alle politiche e alle politiche chiave. IAM Le sovvenzioni vengono spesso utilizzate per autorizzazioni temporanee perché è possibile crearne una, utilizzarne le autorizzazioni ed eliminarla senza modificare le politiche o le politiche chiave. IAM

Le sovvenzioni vengono 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 sull'utilizzo delle sovvenzioni da parte AWS dei servizi, 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 si crea una concessione per una KMS chiave, la concessione consente al committente beneficiario

di richiamare le operazioni di concessione specificate sulla KMS chiave, a condizione che tutte le condizioni specificate nella concessione siano soddisfatte.

- Ogni concessione consente l'accesso a una sola KMS chiave. È possibile creare una concessione per una KMS chiave in un'altra Account AWS.
- Una concessione può consentire l'accesso a una KMS chiave, ma non negare l'accesso.
- Ogni concessione ha un beneficiario principale. Il beneficiario principale può rappresentare una o più identità nella Account AWS stessa KMS chiave o in un account diverso.
- Una concessione può consentire solo operazioni di concessione. Le operazioni di sovvenzione devono essere supportate dalla KMS chiave contenuta nella sovvenzione. Se si specifica un'operazione non supportata, la CreateGrantrichiesta ha esito negativo con un'`ValidationError`eccezione.
- Il beneficiario può utilizzare le autorizzazioni concesse dalla concessione senza specificare la concessione, proprio come farebbe se le autorizzazioni provenissero da una politica o una politica chiave. IAM Tuttavia, poiché AWS KMS API segue un eventuale modello di coerenza, quando si crea, si ritira o si revoca una sovvenzione, è possibile che si verifichi 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 sovvenzioni per ogni chiave. KMS Per informazioni dettagliate, consultare Sovvenzioni per KMS chiave: 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 sovvenzioni per una KMS chiave (`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 sovvenzioni per consentire ai responsabili di visualizzare la KMS chiave, utilizzarla nelle operazioni crittografiche e creare e ritirare le sovvenzioni. 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.

- I mandanti possono ottenere il permesso di creare sovvenzioni sulla base di una politica o di una politica chiave. IAM I mandanti che ottengono `kms:CreateGrant` l'autorizzazione da una politica possono creare sovvenzioni per qualsiasi 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 RetireGrantor RevokeGrant.

Operazioni di concessione

Le AWS KMS operazioni che puoi consentire in una sovvenzione. Se si specificano altre operazioni, la CreateGrantrichiesta 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 consentite devono essere supportate dalla KMS chiave contenuta nella concessione. Se si specifica un'operazione non supportata, la CreateGrantrichiesta ha esito negativo con un'`ValidationError`eccezione. Ad esempio, le concessioni per le KMS chiavi di crittografia simmetriche non possono consentire le operazioni Sign, Verify o. `GenerateMacVerifyMac` Le sovvenzioni per chiavi asimmetriche non possono consentire alcuna operazione che generi KMS chiavi di dati o coppie di chiavi di dati.

Concessione di token

AWS KMS APISegue 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. È possibile 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'GenerateDataKeyoperazione. Utilizza un token di concessione per rappresentare la concessione che concede al chiamante (il titolare del beneficiario) l'autorizzazione a richiamare la chiave `GenerateDataKey` specificata. KMS

```
$ 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 committente uscente può utilizzare un token di concessione in una chiamata all'operazione. RetireGrant

```
$ 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 ListRetirableGrantsoperazioni ListGrantsand 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 IAMutente, un ruolo, un IAMruolo o un utente federato o un utente con ruolo assunto. Il titolare del beneficiario può trovarsi nello stesso account della KMS chiave o in un account diverso. Tuttavia, il committente non può essere un responsabile del servizio, un IAMgruppo o un'organizzazione.AWS

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

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 responsabile uscente può essere qualsiasi AWS principale, inclusi IAM utenti Account AWS, IAM ruoli, utenti federati e utenti con ruolo assunto. Il mandante uscente può avere lo stesso account della KMS chiave o un account diverso.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

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 preside Account AWS in pensione può delegare l'autorizzazione a ritirare una sovvenzione a un IAM preside della stessa. 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)

AWS KMS APISegue 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, consulta Utilizzo di un token di concessione.
- L'CreateGrantoperazione 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 uno specifico beneficiario principale, ad esempio un IAM ruolo, e concedi al beneficiario il permesso principale di utilizzare solo le API operazioni necessarie.
- Utilizzate i vincoli di concessione del contesto di crittografia per garantire che i chiamanti utilizzino la chiave per lo scopo previsto. KMS 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 EncryptionContextEqualconcessione ogni volta che è possibile. Il vincolo di EncryptionContextSubsetconcessione 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 sovvenzioni duplicate hanno la stessa chiaveARN, le stesse API azioni, il principale del beneficiario, il 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 concessioni duplicate, usa

l'operazione. ListGrants Se crei accidentalmente una concessione duplicata, ritirala o revocala il 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 le sovvenzioni nelle politiche, IAM nelle politiche e nelle sovvenzioni chiave. I principali che ottengono l'autorizzazione `CreateGrant` da una concessione hanno autorizzazioni di concessione più limitate.

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

Quando si utilizza una politica o una IAM politica chiave per controllare l'accesso alle operazioni di creazione e gestione delle sovvenzioni, è possibile 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 chiavi di condizione.

km: GrantConstraintType

Consente ai principali di creare una concessione solo quando la concessione include il vincolo di concessione specificato.

km: GrantIsFor 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 sovvenzione, chiama l'CreateGrantoperazione. Specificate una KMS chiave, un titolare del beneficiario 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 `CreateGrant` comando seguente crea una concessione che consente agli utenti autorizzati ad assumere il `keyUserRole` ruolo di richiamare l'operazione Decrypt sulla chiave simmetrica specificata. KMS 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 \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

Se il codice ritenta l'`CreateGrant`operazione o utilizza un metodo AWS SDKche riprova automaticamente le richieste, utilizza il parametro opzionale Name 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. Utilizzare `CreateGrant` con un AWS SDK o 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 sovvenzione sostituiscono le chiavi condizionali in una politica o in una politica chiave. 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 del contesto di crittografia sono validi solo in una concessione per una chiave di crittografia simmetrica. KMS Le operazioni crittografiche con altre KMS chiavi 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 del contesto di crittografia in una concessione per una KMS chiave 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 KMS chiave, 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 `"Deparment": "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 autorizzano loro l'uso KMS delle chiavi 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 KMS chiave specificata solo per una particolare tabella del tuo. Account AWS

Per ottenere questo risultato, esegui l'ListGrantsoperazione 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"
                }
            },
        }
    ]
}
```

Il titolare del beneficiario,exampleUser, può utilizzare questa autorizzazione per creare una concessione che includa qualsiasi sottoinsieme delle operazioni specificate nella concessione originale, ad esempio e. `CreateGrant` `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.

IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

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 il 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'ListGrantsoperazione. È necessario specificare la KMS chiave a cui si applicano le sovvenzioni. È inoltre possibile filtrare l'elenco delle concessioni in base all'ID concessione o all'assegnatario principale. Per ulteriori esempi, consulta Utilizzare `ListGrants` con un AWS SDK o 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

AWS KMS APIDi seguito è riportato 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'CreateGrantoperazione.

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 GenerateDataKeye 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-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 RevokeGrantoperazioni RetireGrante 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 e IAMpolicy chiave. RevokeGrantAPIPuò essere richiamato 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. Questo design consente di controllare il ciclo di vita di una sovvenzione senza modificare le politiche o IAM le politiche chiave. 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.

Esiste un'`kms:RetireGrant`autorizzazione che può essere utilizzata nelle IAM politiche, ma ha 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'`kms:RetireGrant`autorizzazione 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'`Deny`azione con l'`kms:RetireGrant`autorizzazione nelle tue IAM politiche.
- Il Account AWS proprietario della KMS chiave può delegare l'`kms:RetireGrant`autorizzazione a un IAM responsabile dell'account.
- Se il titolare uscente è diverso Account AWS, gli amministratori dell'altro account possono delegare l'autorizzazione `kms:RetireGrant` a ritirare la sovvenzione a un IAM responsabile di quell'account.

AWS KMS APIDi seguito è riportato 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,

usa 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 IAMpolitiche 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 che una dichiarazione politica controlli l'accesso solo quando in una API richiesta viene visualizzato un valore specifico.

Per specificare le condizioni, si utilizzano le chiavi di condizione nell'`Condition`elemento di una dichiarazione di politica con operatori di IAM condizione. Alcune chiavi di condizione si applicano generalmente a AWS, altre sono specifiche a AWS KMS.

I valori delle chiavi condizionali devono rispettare i caratteri e le regole di codifica delle politiche e IAM delle politiche AWS KMS chiave. Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione Formato della policy della chiave. Per i dettagli sulle regole IAM dei documenti politici, consulta i requisiti relativi ai IAM nomi nella Guida per l'IAMutente.

Argomenti

- AWS chiavi di condizione globali
- AWS KMS chiavi di condizione
- AWS KMS chiavi di condizione per AWS Nitro Enclaves

AWS chiavi di condizione globali

AWS definisce le chiavi di condizione globali, un insieme di chiavi di condizioni politiche per tutti i AWS servizi utilizzati IAM per il controllo degli accessi. AWS KMS supporta tutte le chiavi di condizione globali. È possibile utilizzarli nelle politiche e nelle IAM politiche AWS KMS chiave.

Ad esempio, puoi utilizzare la chiave aws: PrincipalArn global condition per consentire l'accesso a una AWS KMS key (KMSchiave) solo quando il principale nella richiesta è rappresentato da 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 IAM policy per consentire l'accesso alle KMS chiavi con un tag particolare.

Per evitare che un AWS servizio venga utilizzato come sostituto confuso in una politica in cui il principale è il responsabile del AWS servizio, puoi utilizzare il aws:SourceArn o aws:SourceAccounttasti di condizione globali. Per informazioni dettagliate, consultare 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 Guida per l'IAMutente. Per esempi di utilizzo delle chiavi di condizione globali nelle IAM politiche, vedere Controlling Access to Requests e Controlling Tag Keys nella Guida per l'IAMutente.

I seguenti argomenti forniscono indicazioni speciali per l'utilizzo delle chiavi di condizione basate su indirizzi IP ed VPC endpoint.

Argomenti

- Utilizzo della condizione con indirizzo IP nelle policy con autorizzazioni AWS KMS
- Utilizzo delle condizioni VPC degli endpoint nelle politiche con autorizzazioni 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. A un'identità si allega una politica come quella mostrata in AWS: Denies Access to AWS Based on the Source IP. IAM Imposti il valore della chiave di condizione `aws:SourceIp` sull'intervallo di indirizzi IP per l'azienda dell'utente. Questa IAM identità ha altre politiche allegate che le consentono di utilizzare Amazon EBSEC2, Amazon e AWS KMS.
2. L'identità tenta di collegare un EBS volume crittografato a un'EC2istanza. 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 AmazonEC2. 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 EBS volume.

Inoltre, la chiave di `aws:sourceIP` condizione non è efficace quando la richiesta proviene da un VPCendpoint Amazon. Per limitare le richieste a un VPC endpoint, incluso un AWS KMS VPCendpoint, usa le chiavi di condizione `aws:sourceVpce` o`aws:sourceVpc`. Per ulteriori informazioni, consulta VPCEndpoints - Controlling the Use of Endpoints nella Amazon VPC User Guide.

Utilizzo delle condizioni VPC degli endpoint nelle politiche con autorizzazioni AWS KMS

AWS KMS supporta gli endpoint Amazon Virtual Private Cloud (AmazonVPC) alimentati da AWS PrivateLink. Puoi utilizzare le seguenti chiavi di condizione globali nelle politiche e IAM nelle politiche chiave per controllare l'accesso alle AWS KMS risorse quando la richiesta proviene da un VPC endpoint VPC o lo utilizza. Per informazioni dettagliate, consultare Usa VPC gli endpoint per controllare l'accesso alle risorse AWS KMS.

- `aws:SourceVpc`limita l'accesso alle richieste provenienti da quanto specificatoVPC.
- `aws:SourceVpce`limita l'accesso alle richieste dall'VPCendpoint specificato.

Se utilizzi questi tasti condizionali per controllare l'accesso alle KMS chiavi, potresti inavvertitamente negare l'accesso ai AWS servizi che utilizzi AWS KMS per tuo conto.

Fai attenzione a evitare una situazione come quella illustrata nell'esempio delle chiavi di condizione con indirizzo IP. Se limiti le richieste di una KMS chiave a un VPC endpoint VPC or, le chiamate AWS KMS da un servizio integrato, come Amazon S3 o EBS Amazon, potrebbero non riuscire. Ciò può accadere anche se la richiesta di origine proviene in ultima analisi dall'endpoint VPC o dall'endpoint. VPC

AWS KMS chiavi di condizione

AWS KMS fornisce un set di chiavi di condizione che è possibile utilizzare nelle politiche e nelle IAM politiche chiave. Queste chiavi di condizione sono specifiche per AWS KMS. Ad esempio, è possibile utilizzare la chiave di `kms:EncryptionContext:`*`context-key`* condizione per richiedere un particolare contesto di crittografia quando si controlla l'accesso a una chiave di crittografia KMS simmetrica.

Condizioni per una richiesta di operazione API

Molte chiavi AWS KMS condizionali controllano l'accesso a una KMS chiave 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 IAM politica per consentire l'uso dell'CreateKeyoperazione 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 KMS le chiavi utilizzate nelle operazioni API

Alcune chiavi AWS KMS condizionali possono controllare l'accesso alle operazioni in base a una proprietà della KMS chiave utilizzata nell'operazione. Ad esempio, puoi utilizzare la KeyOrigin condizione kms: per consentire ai principali di richiamare GenerateDataKeyuna KMS chiave solo quando `Origin` la KMS chiave 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 relativa a una risorsa KMS chiave, ovvero un'operazione autorizzata per una chiave particolareKMS. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'operazione. Se si utilizza questo tipo di chiave condizionale con un'operazione non autorizzata per una particolare risorsa KMS chiave, ad esempio ListKeys, l'autorizzazione non è valida perché la condizione non può mai essere soddisfatta. Non vi è alcuna risorsa KMS chiave coinvolta nell'autorizzazione dell'`ListKeys`operazione e nessuna `KeySpec` proprietà.

I seguenti argomenti 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. IAMdefinisce due operatori di set `ForAnyValue` e`ForAllValues`, a tal fine. 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, consulta 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 API chiamata 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 KMS chiave 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 sugli operatori `ForAnyValue` e `ForAllValues` set, consultate Utilizzo di più chiavi e valori nella Guida per l'IAMutente. Per informazioni sul rischio di utilizzare l'operatore *`ForAllValues`* set con una condizione a valore singolo, vedere Avviso di sicurezza: ForAllValues con chiave a valore singolo nella Guida per l'IAMutente.

Argomenti

- kms:BypassPolicyLockoutSafetyCheck
- kms:CallerAccount

- kms: CustomerMasterKeySpec (obsoleto)
- 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 policy
`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 PutKeyPolicyoperazioni CreateKeyand in base al valore del `BypassPolicyLockoutSafetyCheck` parametro nella richiesta.

L'esempio seguente di dichiarazione IAM politica impedisce agli utenti di aggirare il controllo di sicurezza del blocco delle politiche negando loro l'autorizzazione a creare KMS chiavi quando il valore del `BypassPolicyLockoutSafetyCheck` parametro nella richiesta è `CreateKey` `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`. L'esempio

seguente di dichiarazione politica derivante da una politica chiave impedisce agli utenti di aggirare il controllo di sicurezza relativo al blocco delle politiche quando modificano la politica 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 tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy
`kms:CallerAccount`	Stringa	A valore singolo	KMSoperazioni sulle risorse chiave Operazioni dell'archivio delle	Policy delle chiavi e policy IAM

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy
			chiavi personalizzate	

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. Ma 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 relativa alle risorse KMS chiave, ovvero a qualsiasi AWS KMS operazione che utilizza una chiave particolareKMS. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'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 di AmazonEBS. Combina un `Principal` elemento che specifica tutte le AWS identità con la chiave `kms:CallerAccount` condizionale per consentire efficacemente 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 autorizzando solo le richieste che arrivano tramite AmazonEBS. 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: CustomerMasterKeySpec (obsoleto)

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 policy
`kms:DataKeyPairSpec`	Stringa	A valore singolo	`GenerateDataKeyPair` `GenerateDataKeyPairWithoutPlaintext`	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso alle GenerateDataKeyPairWithoutPlaintextoperazioni GenerateDataKeyPairand 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.

L'esempio seguente di dichiarazione sulla politica chiave utilizza la chiave `kms:DataKeyPairSpec` condition per consentire agli utenti di utilizzare la KMS chiave per generare solo coppie di chiavi di RSA dati.

```
{
  "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 policy

`kms:EncryptionAlgorithm`	Stringa	A valore singolo	`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Policy delle chiavi e policy IAM

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 le ReEncryptoperazioni Encrypt, Decrypt and, controlla l'accesso in base al valore del EncryptionAlgorithmparametro 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 effetto sulle operazioni eseguite all'esterno AWS KMS, come la crittografia con la chiave pubblica in una KMS coppia di chiavi 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 KMS chiave, utilizzate una dichiarazione politica con un `Deny` effetto e un operatore di `StringNotEquals` condizione. Ad esempio, la seguente dichiarazione sulla politica chiave vieta ai responsabili che possono assumere il `ExampleRole` ruolo di utilizzare questa KMS chiave nelle operazioni crittografiche specificate, a meno che l'algoritmo di crittografia contenuto nella richiesta non sia `RSAES_OAEP_SHA_256` un algoritmo di crittografia asimmetrico utilizzato con le chiavi. RSA KMS

A differenza di un'informativa che consente a un utente di utilizzare un particolare algoritmo di crittografia, una dichiarazione politica con una doppia negazione come questa impedisce ad altre politiche e concessioni per questa KMS chiave di consentire a questo ruolo di utilizzare altri algoritmi di crittografia. L'informativa contenuta `Deny` in questa dichiarazione politica chiave ha la precedenza su qualsiasi politica o IAM politica chiave che abbia un `Allow` effetto e ha la precedenza su tutte le concessioni relative a questa chiave e ai suoi principi. KMS

```
{
  "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 solo chiavi di crittografia simmetriche e l'algoritmo. KMS `SYMMETRIC_DEFAULT`

Ad esempio, questa policy IAM limita le sue entità principali alla crittografia simmetrica. Nega l'accesso a qualsiasi KMS chiave nell'account di esempio per le operazioni crittografiche a meno che l'algoritmo di crittografia specificato nella richiesta o utilizzato nell'operazione non sia _. SYMMETRIC DEFAULT Include `GenerateDataKey*` aggiunte GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair, e GenerateDataKeyPairWithoutPlaintextalle 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 condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy

`kms:EncryptionContext: context-key`	Stringa	A valore singolo	`CreateGrant`
			`Encrypt`
			`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
			`RetireGrant`
				Policy delle chiavi e policy IAM

È possibile utilizzare la chiave `kms:EncryptionContext:context-key` condition per controllare l'accesso a una KMSchiave di crittografia simmetrica basata sul contesto di crittografia in una richiesta di operazione crittografica. 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: condition. EncryptionContextKeys

> **Note**
>
> I valori delle chiavi condizionali devono essere conformi alle regole dei caratteri per le politiche e IAM le politiche chiave. 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 IAM dei documenti relativi alle policy, consultate i requisiti relativi ai IAM nomi nella Guida per l'IAMutente.

Non è possibile specificare un contesto di crittografia in un'operazione crittografica con una chiave asimmetrica o una KMS chiave. HMAC KMS Gli algoritmi e gli algoritmi asimmetrici non supportano un contesto di crittografiaMAC.

Per utilizzare la chiave di condizione kms:EncryptionContext: context-key, sostituisci il segnaposto con la chiave di contesto di crittografia*`context-key`*. 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 contesti di crittografia in ogni API richiesta, la coppia di contesto di crittografia con la chiave di contesto 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` or `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`.

L'esempio seguente di dichiarazione sulla politica chiave consente ai responsabili che possono assumere il ruolo di utilizzare la KMS chiave in una `GenerateDataKey` richiesta 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, consultare 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: la chiave contestuale e `kms:EncryptionContextKeys`nella dichiarazione 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 `ForAllValuese` 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 i dettagli sull'operatore *`ForAllValues`* set, consulta Using multiple keys and values nella Guida per l'IAMutente.

```
{
  "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 KMS chiave per una particolare operazione. L'esempio seguente di dichiarazione sulla politica delle chiavi utilizza

un `Deny` effetto per vietare al principale di utilizzare la KMS chiave se il contesto di crittografia nella richiesta include una coppia di contesti di `Stage=Restricted` crittografia. 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 i dettagli sulla logica utilizzata per interpretare queste condizioni, vedere Creazione di una condizione con più chiavi o valori nella Guida per l'IAMutente.

> **Note**
>
> Le versioni precedenti di questo argomento mostravano dichiarazioni politiche 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:Departmentekms:EncryptionContext:Project`), sono connesse implicitamente 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, utilizzate entrambe kms:EncryptionContext: la chiave contestuale 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. L'esempio seguente di dichiarazione sulla politica delle chiavi utilizza un `Deny` effetto per vietare al principale di utilizzare KMS le chiavi se il contesto di crittografia nella richiesta include un `Stage=Restricted` o .pair. `Stage=Production`

Valori multipli (`Restricted` e `Production`) per la stessa chiave (`kms:EncryptionContext:Stage`) sono implicitamente collegati da un'OR. Per i dettagli, consulta Logica di valutazione per condizioni con più chiavi o valori nella Guida per l'IAMutente.

```
{
  "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 kms: EncryptionContextKeys policy 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 `kms:EncryptionContext:context-key` condizione può essere una variabile politica. 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.

Si consideri una situazione in cui si desidera che tutti gli utenti che possono assumere un ruolo utilizzino la stessa KMS chiave per crittografare e decrittografare i propri 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 dichiarazione politica autorizza il `TestTeam` ruolo a crittografare e decrittografare 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}"
    }
  }
}
```

È possibile utilizzare una variabile di IAM policy solo nel valore della chiave 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 informazioni dettagliate sull'identificazione di utenti e ruoli in Amazon Cognito, consulta IAMRuoli nella Amazon Cognito Developer Guide.

```
{
  "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 policy

`kms:EncryptionContextKeys`	Stringa (elenco)	Multivalore	`CreateGrant`
			`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
				Policy delle chiavi e policy IAM

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy

			`GenerateDataKeyWithoutPlaintext`
			
			`ReEncrypt`
			
			`RetireGrant`	

È possibile utilizzare la chiave `kms:EncryptionContextKeys` condition per controllare l'accesso a una KMSchiave di crittografia simmetrica basata sul contesto di crittografia in una richiesta di operazione crittografica. 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 crittografica con una chiave asimmetrica KMS o una chiave. HMAC KMS Gli algoritmi e gli algoritmi asimmetrici non supportano un contesto di crittografiaMAC.

> **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 chiave. AWS KMS 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 dei documenti relativi alle IAM policy, consultate i requisiti relativi ai IAM nomi nella Guida per l'IAMutente.

Si tratta di una chiave di condizione multivalore. È possibile specificare più coppie di contesti di crittografia in ogni API richiesta. `kms:EncryptionContextKeys`confronta le chiavi del contesto di crittografia nella richiesta con l'insieme di chiavi del contesto di crittografia nella politica. Per determinare il modo in cui questi set vengono confrontati, devi fornire un operatore `ForAnyValue` o

`ForAllValues` nella condizione di policy. Per i dettagli sugli operatori del set, vedere Utilizzo di più chiavi e valori nella Guida per l'IAMutente.

- `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 dichiarazione politica consente l'uso di una KMS chiave per le operazioni specificate, ma solo quando almeno una delle coppie di contesti di crittografia nella richiesta include la `AppName` chiave, 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 StringEqualscondizione 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"
    }
  }
}
```

È inoltre possibile utilizzare la chiave `kms:EncryptionContextKeys` condizionale per richiedere un contesto di crittografia (qualsiasi contesto di crittografia) nelle operazioni crittografiche che utilizzano la KMS chiave;.

L'esempio seguente di dichiarazione di policy chiave utilizza la chiave `kms:EncryptionContextKeys` condition con l'operatore di condizione Null per consentire l'accesso a una KMS chiave solo quando il contesto di crittografia nella API richiesta non è nullo. 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 policy
`kms:ExpirationModel`	Stringa	A valore singolo	`ImportKeyMaterial`	Policy delle chiavi e policy IAM

Il tasto `kms:ExpirationModel` condition controlla l'accesso all'ImportKeyMaterialoperazione in base al valore del ExpirationModelparametro 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 ValidToparametro. 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 dichiarazione politica di esempio utilizza la chiave `kms:ExpirationModel` condition per consentire agli utenti di importare materiale chiave in una KMS chiave solo quando la richiesta include il `ExpirationModel` parametro 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 policy
`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'CreateGrantoperazione in base al tipo di vincolo di concessione 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 policy

`kms:GrantIsForAWSResource`	Booleano	A valore singolo	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				Policy delle chiavi e policy IAM

Consente o nega l'CreateGrantautorizzazione per RevokeGrantle 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 AmazonEBS, di creare sovvenzioni su questa KMS chiave 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 policy
`kms:GrantOperations`	Stringa	Multivalore	`CreateGrant`	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'CreateGrantoperazione 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 i dettagli sugli operatori di set, vedere Utilizzo di più chiavi e valori nella Guida per l'IAMutente.

- `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 inclusa 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 policy
`kms:GranteePrincipal`	Stringa	A valore singolo	`CreateGrant`	Policy IAM e della chiave

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'CreateGrantoperazione in base al valore del GranteePrincipalparametro nella richiesta. Ad esempio, è possibile creare concessioni per utilizzare una KMS chiave solo quando il principale beneficiario nella `CreateGrant` richiesta corrisponde al capitale specificato nell'istruzione condizionale.

Per specificare il principale beneficiario, usa il nome Amazon Resource Name (ARN) di un AWS principale. I principali validi includono IAM utenti Account AWS, IAM ruoli, utenti federati e utenti assunti. Per informazioni sulla ARN sintassi di un principale, consulta la Guida per l'IAMARNsIAMutente.

L'esempio seguente di dichiarazione politica chiave utilizza la chiave `kms:GranteePrincipal` condition per creare sovvenzioni per una KMS chiave solo quando il beneficiario principale della sovvenzione è il. `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 policy
`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'DeriveSharedSecretoperazione in base al valore del `KeyAgreementAlgorithm` parametro nella richiesta. L'unico valore valido per `KeyAgreementAlgorithm` è`ECDH`.

Ad esempio, la seguente dichiarazione di politica chiave utilizza la chiave `kms:KeyAgreementAlgorithm` condition per negare qualsiasi accesso a DeriveSharedSecret meno che non lo `KeyAgreementAlgorithm` sia`ECDH`.

```
{
       "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 policy
`kms:KeyOrigin`	Stringa	A valore singolo	`CreateKey` KMSoperazioni sulle risorse chiave	Policy IAM Policy delle chiavi e policy IAM

Il tasto `kms:KeyOrigin` condition controlla l'accesso alle operazioni in base al valore della `Origin` proprietà della KMS chiave 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'CreateKeyoperazione 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 KMS chiave 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 un'origine esterna (`EXTERNAL`).

L'esempio seguente di dichiarazione politica chiave utilizza la chiave `kms:KeyOrigin` condition per creare una KMS chiave solo quando AWS KMS crea il materiale chiave.

```
{
  "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"
        }
      }
    }
  ]
}
```

È inoltre possibile utilizzare il tasto `kms:KeyOrigin` condition per controllare l'accesso alle operazioni che utilizzano o gestiscono una KMS chiave in base alla `Origin` proprietà della KMS chiave utilizzata per l'operazione. L'operazione deve essere un'operazione relativa a una risorsa KMS chiave, ovvero un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'operazione.

Ad esempio, la seguente IAM politica consente ai responsabili di eseguire le operazioni sulle risorse KMS chiave specificate, ma solo con KMS le chiavi dell'account che sono state create in un archivio di 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 policy
`kms:KeySpec`	Stringa	A valore singolo	`CreateKey` KMSoperazioni sulle risorse chiave	Policy IAM Policy delle chiavi e policy IAM

Il tasto `kms:KeySpec` condition controlla l'accesso alle operazioni in base al valore della `KeySpec` proprietà della KMS chiave creata o utilizzata nell'operazione.

È possibile utilizzare questa chiave di condizione in una IAM politica per controllare l'accesso all'CreateKeyoperazione in base al valore del KeySpecparametro in una `CreateKey` richiesta. Ad esempio, è possibile utilizzare questa condizione per consentire agli utenti di creare solo chiavi di crittografia simmetriche o solo KMS HMAC KMS chiavi.

L'esempio seguente di dichiarazione IAM politica utilizza la chiave `kms:KeySpec` condition per consentire ai principali di creare solo RSA chiavi asimmetriche. KMS L'autorizzazione è valida solo quando `KeySpec` nella richiesta inizia con `RSA_`.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

È inoltre possibile utilizzare il tasto `kms:KeySpec` condition per controllare l'accesso alle operazioni che utilizzano o gestiscono una KMS chiave in base alla `KeySpec` proprietà della KMS chiave utilizzata per l'operazione. L'operazione deve essere un'operazione relativa a una risorsa KMS chiave, ovvero un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'operazione.

Ad esempio, la seguente IAM politica consente ai responsabili di eseguire le operazioni sulle risorse KMS chiave specificate, ma solo con KMS chiavi di crittografia simmetriche 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: CustomerMasterKeySpec (obsoleto)
- 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 policy
`kms:KeyUsage`	Stringa	A valore singolo	`CreateKey` KMSoperazioni sulle risorse chiave	Policy IAM Policy delle chiavi e policy IAM

Il tasto `kms:KeyUsage` condition controlla l'accesso alle operazioni in base al valore della `KeyUsage` proprietà della KMS chiave creata o utilizzata nell'operazione.

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'CreateKeyoperazione in base al valore del KeyUsageparametro nella richiesta. I valori validi per `KeyUsage` sono `ENCRYPT_DECRYPTSIGN_VERIFY`,`GENERATE_VERIFY_MAC`, e`KEY_AGREEMENT`.

Ad esempio, è possibile creare una KMS chiave solo quando `KeyUsage` è `ENCRYPT_DECRYPT` o negare l'autorizzazione di un utente quando `KeyUsage` è`SIGN_VERIFY`.

L'esempio seguente di dichiarazione IAM politica utilizza la chiave `kms:KeyUsage` condition per creare una KMS chiave solo quando `KeyUsage` è`ENCRYPT_DECRYPT`.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "ENCRYPT_DECRYPT"
    }
  }
}
```

È inoltre possibile utilizzare il tasto `kms:KeyUsage` condition per controllare l'accesso alle operazioni che utilizzano o gestiscono una KMS chiave in base alla `KeyUsage` proprietà della KMS chiave nell'operazione. L'operazione deve essere un'operazione relativa a una risorsa KMS chiave, ovvero un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'operazione.

Ad esempio, la seguente IAM politica consente ai responsabili di eseguire le operazioni sulle risorse KMS chiave specificate, ma solo con KMS le chiavi dell'account 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 policy
`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 VerifyMacoperazioni GenerateMacand in base al valore del `MacAlgorithm` parametro nella richiesta.

L'esempio seguente di policy chiave consente agli utenti che possono assumere il `testers` ruolo di utilizzare la HMAC KMS chiave per generare e verificare i HMAC tag solo quando l'MACalgoritmo 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 specifica più di un MAC algoritmo in una singola dichiarazione di condizione, la condizione richiede entrambi gli algoritmi, anziché l'uno o l'altro.

```
{
  "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 policy
`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 dichiarazione politica chiave utilizza la chiave `kms:MessageType` condition per utilizzare una KMS chiave asimmetrica per firmare un messaggio, ma non un message digest.

```
{
  "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 policy
`kms:MultiRegion`	Booleano	A valore singolo	`CreateKey` KMSoperazioni sulle risorse chiave	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 KMS chiavi e all'CreateKeyoperazione in base al valore della `MultiRegion` proprietà della KMS chiave. I valori validi sono `true` (multiregione) e `false` (singola Regione). Tutte KMS le chiavi hanno una `MultiRegion` proprietà.

Ad esempio, la seguente dichiarazione IAM politica utilizza la chiave `kms:MultiRegion` condition per consentire ai principali di creare solo chiavi a regione singola.

```
{
  "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 policy
`kms:MultiRegionKeyType`	Stringa	A valore singolo	`CreateKey` KMSoperazioni sulle risorse chiave	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 KMS chiavi e l'CreateKeyoperazione in base alla `MultiRegionKeyType` proprietà della KMS chiave. I valori validi sono `PRIMARY` e `REPLICA`. Solo le chiavi multiregione hanno una proprietà `MultiRegionKeyType`.

In genere, si utilizza la chiave di `kms:MultiRegionKeyType` condizione in una IAM politica per controllare l'accesso a più KMS chiavi. 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 dichiarazione di IAM politica utilizza la chiave `kms:MultiRegionKeyType` condition per consentire ai responsabili di pianificare e annullare l'eliminazione delle chiavi solo per le chiavi di replica multiregionali nel territorio specificato. Account AWS

```
{
  "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 chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy
`kms:PrimaryRegion`	Stringa (elenco)	A valore singolo	`UpdatePrimaryRegion`	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare le regioni di destinazione in un'UpdatePrimaryRegionoperazione. Queste sono quelle Regioni AWS che possono ospitare le tue chiavi primarie multiregionali.

Il tasto `kms:PrimaryRegion` condition controlla l'accesso all'UpdatePrimaryRegionoperazione 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 policy
`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'ReEncryptoperazione a seconda che la richiesta specifichi una KMS chiave di destinazione uguale a quella utilizzata per la crittografia originale.

Ad esempio, la seguente dichiarazione sulla politica di chiave utilizza la chiave `kms:ReEncryptOnSameKey` condition per ricrittografare solo quando la KMS chiave di destinazione è la stessa 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 policy
`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 alias particolare per identificare la KMS chiave. La chiave `kms:RequestAlias` condizionale controlla l'accesso a una KMS chiave utilizzata in un'operazione crittografica o in `DescribeKey` base all'alias che identifica quella KMS chiave nella richiesta. `GetPublicKey` (Questa condizione politica non ha alcun effetto sull'GenerateRandomoperazione perché l'operazione non utilizza una KMS chiave o un alias.)

Questa condizione supporta il controllo degli accessi basato sugli attributi (ABAC) in AWS KMS, che consente di controllare l'accesso alle KMS chiavi 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 KMS chiave senza modificare le politiche o le concessioni. Per informazioni dettagliate, consultare ABACper 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 alias ARN nel valore di questa chiave di condizione.

Per soddisfare questa condizione, il valore del `KeyId` parametro nella richiesta deve essere un nome alias o un alias corrispondente. ARN Se la richiesta utilizza un identificatore di chiave diverso, 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'GenerateDataKeyoperazione sulla chiave. KMS Tuttavia, ciò è consentito solo quando il valore del `KeyId` parametro nella richiesta è `alias/finance-key` o è un alias ARN 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 o. CreateAliasDeleteAlias 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 policy
`kms:ResourceAliases`	Stringa (elenco)	Multivalore	KMSoperazioni sulle risorse chiave	Solo policy IAM

Utilizzate questa chiave di condizione per controllare l'accesso a una KMS chiave in base agli alias associati alla KMS chiave. L'operazione deve essere un'operazione relativa a una risorsa KMS chiave, ovvero un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'operazione.

Questa condizione supporta il controllo degli accessi basato sugli attributi () ABAC in. AWS KMS ConABAC, è possibile controllare l'accesso alle KMS chiavi in base ai tag assegnati a una KMS chiave e agli alias associati a una chiave. KMS È possibile utilizzare tag e alias per consentire o negare l'accesso a una KMS chiave senza modificare le politiche o le concessioni. Per informazioni dettagliate, consultare ABACper AWS KMS.

Un alias deve essere univoco in una regione Account AWS and, ma questa condizione consente di controllare l'accesso a più KMS chiavi nella stessa regione (utilizzando l'operatore di `StringLike` confronto) o a più KMS chiavi in diversi Regioni AWS account.

> **Note**
>
> La ResourceAliases condizione kms: è efficace solo quando la KMS chiave è conforme agli alias per quota di chiavi. KMS Se una KMS chiave supera questa quota, ai principali che sono autorizzati a utilizzare la KMS chiave in base alla `kms:ResourceAliases` condizione 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 alias ARN nel valore di questa chiave di condizione. Per soddisfare la condizione, la

KMS chiave utilizzata nell'operazione deve avere l'alias specificato. Non importa se o come la KMS chiave viene identificata nella richiesta dell'operazione.

Si tratta di una chiave di condizione multivalore che confronta l'insieme di alias associati a una KMS chiave con l'insieme di alias della politica. Per determinare il modo in cui questi set vengono confrontati, devi fornire un operatore `ForAnyValue` o `ForAllValues` nella condizione di policy. Per i dettagli sugli operatori del set, consulta Utilizzo di più chiavi e valori nella Guida per l'utente. IAM

- ForAnyValue: almeno un alias associato alla KMS chiave deve corrispondere a un alias nella condizione della policy. Sono consentiti altri alias. Se la KMS chiave non ha alias, la condizione non è soddisfatta.
- ForAllValues: ogni alias associato alla KMS chiave deve corrispondere a un alias nella policy. Questo operatore di set limita gli alias associati alla KMS chiave a quelli presenti nelle condizioni della policy. Non richiede alcun alias, ma vieta alias non specificati.

Ad esempio, la seguente dichiarazione di IAM policy consente al principale di richiamare l'GenerateDataKeyoperazione su qualsiasi KMS chiave dello specificato Account AWS associata all'`finance-key`alias. (Le politiche chiave delle KMS chiavi interessate devono inoltre consentire all'account del principale di utilizzarle per questa operazione.) Per indicare che la condizione è soddisfatta quando uno dei tanti alias che possono essere associati alla KMS chiave lo è`alias/finance-key`, la condizione utilizza l'operatore `ForAnyValue` set.

Poiché la `kms:ResourceAliases` condizione è basata sulla risorsa e non sulla richiesta, una chiamata a `GenerateDataKey` ha esito positivo per qualsiasi KMS chiave associata all'`finance-key`alias, anche se la richiesta utilizza un ID o una chiave ARN 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"
    }
  }
```

```
}
```

La seguente dichiarazione IAM politica di esempio consente al principale di abilitare e disabilitare KMS le chiavi, ma solo quando tutti gli alias delle KMS chiavi includono ".» `Test` Questa dichiarazione politica utilizza due condizioni. La condizione con l'operatore `ForAllValues` set richiede che tutti gli alias associati alla KMS chiave includano «Test». La condizione con l'operatore `ForAnyValue` set richiede che la KMS chiave abbia almeno un alias con «Test». Senza la `ForAnyValue` condizione, questa dichiarazione politica avrebbe consentito al principale di utilizzare KMS chiavi 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 policy
`kms:ReplicaRegion`	Stringa (elenco)	A valore singolo	`ReplicateKey`	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare il numero Regioni AWS in cui un principale può replicare una chiave multiregionale. La chiave `kms:ReplicaRegion` condizionale controlla l'accesso all'ReplicateKeyoperazione in base al valore del ReplicaRegionparametro 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` or`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'ReplicateKeyoperazione 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'ReplicateKeyoperazione. Per controllare l'accesso all'UpdatePrimaryRegionoperazione, 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 policy
`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'CreateGrantoperazione in base al valore del RetiringPrincipalparametro nella richiesta. Ad esempio, è possibile creare concessioni per utilizzare una KMS chiave solo quando il valore `RetiringPrincipal` in nella `CreateGrant` richiesta corrisponde `RetiringPrincipal` a quello dell'istruzione condition.

Per specificare il principale che andrà in pensione, usa il nome Amazon Resource Name (ARN) di un AWS principale. I principali validi includono IAM utenti Account AWS, IAM ruoli, utenti federati e utenti assunti. Per informazioni sulla ARN sintassi di un principale, consulta la Guida per l'IAMARNsIAMutente.

L'esempio seguente di dichiarazione politica chiave consente a un utente di creare concessioni per la KMS chiave. La chiave `kms:RetiringPrincipal` condizionale limita l'autorizzazione alle `CreateGrant` richieste in cui è il principale beneficiario della sovvenzione. `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 policy
`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 principali possono specificare nel `RotationPeriodInDays` parametro di una EnableKeyRotationrichiesta.

`RotationPeriodInDays`specifica 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 policy
`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 principali possono specificare nel `PendingWindowInDays` parametro di una ScheduleKeyDeletionrichiesta.

`PendingWindowInDays`Speciifica 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 policy
`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 SigningAlgorithmparametro 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 KMS chiavi asimmetrica esterna a. AWS KMS

Il seguente esempio di policy chiave consente agli utenti che possono assumere il `testers` ruolo di utilizzare la KMS chiave 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 policy
`kms:ValidTo`	Timestamp	A valore singolo	`ImportKeyMaterial`	Policy delle chiavi e policy IAM

Il tasto `kms:ValidTo` condition controlla l'accesso all'ImportKeyMaterialoperazione in base al valore del ValidToparametro nella richiesta, che determina quando scade il 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 ExpirationModelparametro 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.

L'esempio seguente di dichiarazione politica consente a un utente di importare materiale chiave in una KMS chiave. 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 policy
`kms:ViaService`	Stringa	A valore singolo	KMSoperazioni sulle risorse chiave	Policy delle chiavi e policy IAM

La chiave di `kms:ViaService` condizione limita l'uso di una KMS chiave alle richieste provenienti da AWS servizi specifici. È possibile specificare uno o più servizi in ciascuna chiave di condizione `kms:ViaService`. L'operazione deve essere un'operazione di risorsa KMS chiave, ovvero un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella Azioni e risorse, cerca il valore di `KMS key` nella `Resources` colonna relativa all'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 anche utilizzare una chiave `kms:ViaService` condizionale per negare l'autorizzazione all'uso di una KMS chiave quando la richiesta proviene da servizi particolari. 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 all'uso della KMS chiave. 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 concedere agli utenti l'accesso a un AWS servizio che si integra con AWS KMS, consulta la documentazione relativa al servizio integrato.

Tutte le Chiavi gestite da AWS utilizzano una chiave di condizione `kms:ViaService` nel documento della policy delle chiavi. Questa condizione consente di utilizzare la KMS chiave solo per le richieste provenienti dal servizio che ha creato la KMS chiave. Per visualizzare la politica chiave di an Chiave gestita da AWS, usa l'GetKeyPolicyoperazione.

La chiave di `kms:ViaService` condizione è valida IAM nelle dichiarazioni politiche chiave. 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 dai clienti. 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 servizio	AWS KMS ViaService nome
Operazioni AI di Amazon	`aiops.`*`AWS_region`*` .amazonaws.com`
AWS App Runner	`apprunner.`*`AWS_region`*`.amazonaws.com`

Nome servizio	AWS KMS ViaService nome
AWS AppFabric	`appfabric.` *`AWS_region`* `.amazonaws.com`
Amazon AppFlow	`appflow.`*`AWS_region`* `.amazonaws.com`
AWS Application Migration Service	`mgn.`*`AWS_region`* `.amazonaws.com`
Amazon Athena	`athena.`*`AWS_region`* `.amazonaws.com`
AWS Audit Manager	`auditmanager.` *`AWS_region`* `.amazonaws.com`
Amazon Aurora	`rds.`*`AWS_region`* `.amazonaws.com`
AWS Backup	`backup.`*`AWS_region`* `.amazonaws.com`
AWS Backup Gateway	`backup-gateway.` *`AWS_region`* `.amazonaws.com`
Copia del modello Amazon Bedrock	`bedrock.`*`AWS_region`* `.amazonaws.com`
Amazon Chime SDK	`chimevoiceconnector.` *`AWS_region`* `.amazonaws.com`
AWS Clean Rooms ML	`cleanrooms-ml.` *`AWS_region`* `.amazonaws.com`
AWS CodeArtifact	`codeartifact.` *`AWS_region`* `.amazonaws.com`
CodeGuru Revisore Amazon	`codeguru-reviewer.` *`AWS_region`* `.amazonaws.com`
Amazon Comprehend	`comprehend.` *`AWS_region`* `.amazonaws.com`
Amazon Connect	`connect.`*`AWS_region`* `.amazonaws.com`

Nome servizio	AWS KMS ViaService nome
Customer Profiles Amazon Connect	`profile.`*`AWS_region`*` .amazonaws.com`
Amazon Q in Connect	`wisdom.`*`AWS_region`*` .amazonaws.com`
AWS Database Migration Service (AWS DMS)	`dms.`*`AWS_region`*` .amazonaws.com`
AWS DeepRacer	`deepracer.`*`AWS_region`*` .amazonaws.com`
AWS Directory Service	`directoryservice.`*`AWS_region`*` .amazonaws.com`
Amazon DocumentDB	`docdb-elastic.`*`AWS_region`*` .amazonaws.com`
Amazon DynamoDB	`dynamodb.`*`AWS_region`*` .amazonaws.com`
Amazon EC2 Systems Manager (SSM)	`ssm.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic Block Store (AmazonEBS)	`ec2.`*`AWS_region`*` .amazonaws.com` (solo EBS)
Registro Amazon Elastic Container (AmazonECR)	`ecr.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic File System (AmazonEFS)	`elasticfilesystem.`*`AWS_region`*` .amazonaws.com`
Amazon ElastiCache	Includi entrambi ViaService i nomi nel valore della chiave condizionale: • `elasticache.`*`AWS_region`*` .amazonaws.com` • `dax.`*`AWS_region`*` .amazonaws.com`
AWS Elemental MediaTailor	`mediatailor.`*`AWS_region`*` .amazonaws.com`

Nome servizio	AWS KMS ViaService nome
AWS Risoluzione dell'entità	`entityresolution.` *`AWS_region`* `.amazonaws.com`
Amazon EventBridge	`events.`*`AWS_region`*` .amazonaws.com`
Amazon FinSpace	`finspace.` *`AWS_region`* `.amazonaws.com`
Amazon Forecast	`forecast.` *`AWS_region`* `.amazonaws.com`
Amazon FSx	`fsx.`*`AWS_region`*` .amazonaws.com`
AWS Glue	`glue.`*`AWS_region`*` .amazonaws.com`
AWS Ground Station	`groundstation.` *`AWS_region`* `.amazonaws.com`
Amazon GuardDuty	`malware-protection.` *`AWS_region`* `.amazonaws.com`
AWS HealthLake	`healthlake.` *`AWS_region`* `.amazonaws.com`
AWS IoT SiteWise	`iotsitewise.` *`AWS_region`* `.amazonaws.com`
Amazon Kendra	`kendra.`*`AWS_region`*` .amazonaws.com`
Amazon Keyspaces (per Apache Cassandra)	`cassandra.` *`AWS_region`* `.amazonaws.com`
Amazon Kinesis	`kinesis.`*`AWS_region`*` .amazonaws.com`
Amazon Data Firehose	`firehose.` *`AWS_region`* `.amazonaws.com`

Nome servizio	AWS KMS ViaService nome
Flusso di video Amazon Kinesis	`kinesisvideo.` *`AWS_region`* `.amazonaws.com`
AWS Lambda	`lambda.`*`AWS_region`* `.amazonaws.com`
Amazon Lex	`lex.`*`AWS_region`* `.amazonaws.com`
AWS License Manager	`license-manager.` *`AWS_region`* `.amazonaws.com`
Servizio di posizione Amazon	`geo.`*`AWS_region`* `.amazonaws.com`
Amazon Lookout per le apparecchiature	`lookoutequipment.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout per le metriche	`lookoutmetrics.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout per Vision	`lookoutvision.` *`AWS_region`* `.amazonaws.com`
Amazon Macie	`macie.`*`AWS_region`* `.amazonaws.com`
Modernizzazione del mainframe AWS	`m2.`*`AWS_region`* `.amazonaws.com`
Modernizzazione del mainframe AWS Test delle applicazioni	`apptest.`*`AWS_region`* `.amazonaws.com`
Blockchain gestita da Amazon	`managedblockchain.` *`AWS_region`* `.amazonaws.com`
Streaming gestito da Amazon per Apache Kafka (Amazon) MSK	`kafka.`*`AWS_region`* `.amazonaws.com`
Flussi di lavoro gestiti da Amazon per Apache Airflow () MWAA	`airflow.`*`AWS_region`* `.amazonaws.com`

Nome servizio	AWS KMS ViaService nome
Amazon MemoryDB	memorydb. *AWS_region* .amazonaws.com
Amazon Monitron	monitron. *AWS_region* .amazonaws.com
Amazon MQ	mq.*AWS_region* .amazonaws.com
Amazon Neptune	rds.*AWS_region* .amazonaws.com
Amazon Nimble Studio	nimble.*AWS_region* .amazonaws.com
AWS HealthOmics	omics.*AWS_region* .amazonaws.com
OpenSearch Servizio Amazon	es.*AWS_region* .amazonaws.com , aoss.*AWS_region* .amazonaws.com
Pacchetti OpenSearch personalizzati Amazon	custom-packages. *AWS_region* .amazonaws.com
AWS Proton	proton.*AWS_region* .amazonaws.com
Database Amazon Quantum Ledger (Amazon) QLDB	qldb.*AWS_region* .amazonaws.com
Amazon RDS Performance Insights	rds.*AWS_region* .amazonaws.com
Amazon Redshift	redshift. *AWS_region* .amazonaws.com
Editor di query Amazon Redshift V2	sqlworkbench. *AWS_region* .amazonaws.com
Amazon Redshift Serverless	redshift-serverless. *AWS_region* .amazonaws.com
Amazon Rekognition	rekognition. *AWS_region* .amazonaws.com

Nome servizio	AWS KMS ViaService nome
Amazon Relational Database Service (AmazonRDS)	`rds.AWS_region .amazonaws.com`
Datastore replicato di Amazon	`ards.AWS_region .amazonaws.com`
Amazon SageMaker AI	`sagemaker. AWS_region .amazonaws.com`
AWS Secrets Manager	`secretsmanager. AWS_region .amazonaws.com`
Amazon Security Lake	`securitylake. AWS_region .amazonaws.com`
Servizio e-mail semplice Amazon (AmazonSES)	`ses.AWS_region .amazonaws.com`
Servizio di notifica semplice Amazon (AmazonSNS)	`sns.AWS_region .amazonaws.com`
Servizio Amazon Simple Queue (AmazonSQS)	`sqs.AWS_region .amazonaws.com`
Amazon Simple Storage Service (Amazon S3)	`s3.AWS_region .amazonaws.com`
AWS Snowball	`importexport. AWS_region .amazonaws.com`
AWS Step Functions	`states.AWS_region .amazonaws.com`
AWS Storage Gateway	`storagegateway. AWS_region .amazonaws.com`
Strumento di gestione degli incidenti AWS Systems Manager	`ssm-incidents. AWS_region .amazonaws.com`
Strumento di gestione degli incidenti AWS Systems Manager Contatti	`ssm-contacts. AWS_region .amazonaws.com`

Nome servizio	AWS KMS ViaService nome
Amazon Timestream	`timestream.` *`AWS_region`* `.amazonaws.com`
Amazon Translate	`translate.` *`AWS_region`* `.amazonaws.com`
Accesso verificato da AWS	`verified-access.` *`AWS_region`* `.amazonaws.com`
Amazon WorkMail	`workmail.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces	`workspaces.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces Thin Client	`thinclient.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces Web	`workspaces-web.` *`AWS_region`* `.amazonaws.com`
AWS X-Ray	`xray.`*`AWS_region`* `.amazonaws.com`

kms:WrappingAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di policy
`kms:WrappingAlgorithm`	Stringa	A valore singolo	`GetParametersForImport`	Policy delle chiavi e policy IAM

Questa chiave di condizione controlla l'accesso all'GetParametersForImportoperazione in base al valore del WrappingAlgorithmparametro 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 policy
`kms:WrappingKeySpec`	Stringa	A valore singolo	`GetParametersForImport`	Policy delle chiavi e policy IAM

Questa chiave di condizione controlla l'accesso all'GetParametersForImportoperazione in base al valore del WrappingKeySpecparametro 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 AWS Nitro Enclaves

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. AWS KMS fornisce chiavi di condizione per supportare Nitro Enclaves. AWS Queste chiavi di condizioni sono valide solo per le richieste di Nitro AWS KMS Enclave.

Quando richiami Decrypt,, DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair, o GenerateRandomAPIIe operazioni con il documento di attestazione firmato da un'enclave, 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 Attestazione crittografica per AWS Nitro Enclaves.

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 dell'enclave con i valori di queste chiavi di condizione. AWS KMS

km: 384 RecipientAttestation ImageSha

AWS KMS Chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni di API	Tipo di policy

`kms:RecipientAttestation:ImageSha384`	Stringa	A valore singolo	`Decrypt`
			`DeriveSharedSecret`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Policy delle chiavi e policy IAM

La chiave `kms:RecipientAttestation:ImageSha384` condizionale controlla l'accesso a `Decrypt` `DeriveSharedSecretGenerateDataKey`,`GenerateDataKeyPair`, e `GenerateRandom` con una KMS chiave quando l'immagine digest del documento di attestazione firmato nella richiesta corrisponde al valore nella chiave di condizione. Il `ImageSha384` valore corrisponde a PCR 0 nel documento di attestazione. Questa chiave condizionale è efficace solo

quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato per un' AWS enclave Nitro.

Questo valore è incluso anche negli CloudTraileventi per le richieste alle enclavi Nitro. AWS KMS

Ad esempio, la seguente dichiarazione sulla politica chiave consente al `data-processing` ruolo di utilizzare la KMS chiave per Decrypt, DeriveSharedSecret, GenerateDataKeye le operazioni. GenerateDataKeyPairGenerateRandom La chiave `kms:RecipientAttestation:ImageSha384` condition consente le operazioni solo quando il valore image digest (PCR0) del documento di attestazione nella richiesta corrisponde al valore dell'image 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: < _ID> RecipientAttestation PCR PCR

AWS KMS Chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni di API	Tipo di policy

`kms:RecipientAttestation:PCR<PCR_ID>`	Stringa	A valore singolo	`Decrypt`
			`DeriveSharedSecret`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Policy delle chiavi e policy IAM

La chiave `kms:RecipientAttestation:PCR<PCR_ID>` condizionale controlla l'accesso a `Decrypt` `DeriveSharedSecret``GenerateDataKey`,`GenerateDataKeyPair`, e `GenerateRandom` con una KMS chiave solo quando la configurazione della piattaforma registra (PCRs) dal documento di attestazione firmato nella richiesta corrisponde alla PCRs chiave di condizione. 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 CloudTraileventi che rappresentano le richieste per le enclavi Nitro. AWS KMS

Per specificare un PCR valore, utilizzate il seguente formato. Concatena l'PCRID al nome della chiave della condizione. Il PCR valore deve essere una stringa esadecimale minuscola fino a 96 byte.

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

Ad esempio, la seguente chiave condizionale specifica un valore particolare perPCR1, che corrisponde all'hash del kernel utilizzato per l'enclave e il processo di bootstrap.

```
kms:RecipientAttestation:PCR1:
 "0x1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcdef
```

L'esempio seguente di dichiarazione chiave consente al `data-processing` ruolo di utilizzare la chiave per l'KMSoperazione Decrypt.

La chiave di `kms:RecipientAttestation:PCR` condizione in questa istruzione consente l'operazione solo quando il PCR1 valore nel documento di attestazione firmato nella richiesta corrisponde al `kms:RecipientAttestation:PCR1` valore nella condizione. Utilizzate l'operatore di `StringEqualsIgnoreCase` policy per richiedere un confronto dei valori senza distinzione tra maiuscole e minuscole. 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":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f15
    }
  }
}
```

Autorizzazioni con privilegi minimi

Poiché KMS le tue chiavi 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 politica KMS chiave solo se prevedi di limitare ulteriormente le autorizzazioni con politiche aggiuntive. IAM Se prevedi di gestire le autorizzazioni tramite IAM policy, limita chi ha la possibilità di creare e allegare IAM le policy ai IAM principali e monitora le modifiche alle policy.

Se consenti tutte le azioni (`kms:*`) sia nella policy chiave che nella IAM policy, il mandante 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. È possibile farlo nominando esplicitamente il principale nella politica chiave o limitando i principi a cui è associata la politica. 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 API chiamata ha il tag specificato nella regola di condizione.

Per informazioni su come vengono valutate le dichiarazioni politiche AWS, vedere Logica di valutazione delle politiche nella Guida per l'IAMutente. Ti consigliamo di esaminare questo argomento prima di scrivere le politiche per ridurre la possibilità che la politica abbia effetti indesiderati, ad esempio fornendo l'accesso a dirigenti che non dovrebbero avere accesso.

> **Tip**
>
> Quando testate un'applicazione in un ambiente non di produzione, utilizzate IAMAccess Analyzer per applicare i privilegi minimi alle vostre politiche. IAM

Se utilizzi IAM gli utenti anziché i IAM ruoli, ti consigliamo vivamente di abilitare l' AWS autenticazione a più fattori () per mitigare la vulnerabilità delle credenziali a lungo termine. MFA Puoi usare MFA per fare quanto segue:

- Richiedete agli utenti di convalidare le proprie credenziali MFA prima di eseguire azioni privilegiate, come la pianificazione dell'eliminazione delle chiavi.
- Dividi la proprietà di un account amministratore, della password e del MFA dispositivo tra individui per implementare l'autorizzazione suddivisa.

Ulteriori informazioni

- AWS politiche gestite per le funzioni lavorative
- Tecniche per scrivere politiche sui privilegi minimi IAM

Implementazione di autorizzazioni con privilegio minimo

Quando concedi a un AWS servizio l'autorizzazione a utilizzare una KMS chiave, 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 KMS chiave quando le richieste vengono trasferite tra AWS i servizi.

Per implementare una strategia con privilegi minimi, si consiglia di utilizzare le chiavi di condizione del contesto di AWS KMS crittografia e le chiavi di condizione globali di origine ARN o dell'account di origine.

Utilizzo delle chiavi di condizione del contesto di crittografia

Il modo più efficace per implementare le autorizzazioni con meno privilegi quando si utilizzano le AWS KMS risorse consiste nell'includere kms:EncryptionContext:chiave contestuale o kms:EncryptionContextKeyschiavi di condizione nella politica che consente ai principali di AWS KMS richiamare operazioni crittografiche. 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 delle condizioni del contesto di crittografia solo quando l'azione nella dichiarazione politica è CreateGranto 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, vedere kms:EncryptionContext:chiave contestuale o kms:EncryptionContextKeys). Se si utilizzano questi tasti di condizione per consentire altre operazioni, ad esempio DescribeKey, l'autorizzazione verrà negata.

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 AWS Secrets Manager 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 di una dichiarazione politica chiave è un responsabile del AWS servizio, si consiglia vivamente di utilizzare il aws:SourceArn o aws:SourceAccountchiavi di condizione globali, oltre alla chiave di `kms:EncryptionContext:`*`context-key`* condizione. I valori ARN e dell'account vengono 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 `aws:SourceArn` o `aws:SourceAccount` global condition, imposta il valore sull'Amazon Resource Name (ARN) o sull'account della risorsa da crittografare. Ad esempio, in un'informativa chiave che AWS CloudTrail autorizza a crittografare un percorso, imposta il valore di `aws:SourceArn` to ARN of the trail. Quando possibile, utilizzare `aws:SourceArn`, che è più specifico. Imposta il valore su ARN o su un ARN pattern con caratteri jolly. Se non conosci la natura ARN della risorsa, usa `aws:SourceAccount` invece.

> **Note**
>
> Se una risorsa ARN include caratteri non consentiti in una politica AWS KMS chiave, non è possibile utilizzare quella risorsa ARN nel valore della chiave di `aws:SourceArn` condizione. 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 CloudTrail consente di utilizzare la KMS chiave per generare la chiave di dati che utilizza per crittografare una traccia. Le condizioni `aws:SourceArn` e `kms:EncryptionContext:`*`context-key`* sono valutate in modo indipendente. Qualsiasi richiesta di utilizzo della KMS chiave per l'operazione specificata deve soddisfare entrambe le condizioni.

Per limitare l'autorizzazione del servizio alla `finance` traccia nell'account di esempio (111122223333) e `us-west-2` nella regione, questa dichiarazione politica imposta la chiave ARN di `aws:SourceArn` condizione sulla traccia specifica. L'istruzione condizionale utilizza l'ArnEqualsoperatore per garantire che ogni elemento di 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow CloudTrail to encrypt logs",
      "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/*"
          ]
```

```
        }
      }
    }
  ]
}
```

ABACper 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 consentendo 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 abilitano ABAC AWS KMS forniscono un modo potente e flessibile per autorizzare i responsabili all'uso delle KMS chiavi senza modificare le politiche o gestire le concessioni. Ma occorre usare queste funzionalità con cura in modo che ai principali non sia inavvertitamente autorizzato o negato l'accesso.

Se utilizzateABAC, tenete presente che l'autorizzazione a gestire tag e alias è ora un'autorizzazione di controllo dell'accesso. Assicurati di conoscere i tag e gli alias esistenti su tutte le KMS chiavi prima di implementare una politica che dipenda 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 può consentire o negare l'autorizzazione a una chiave. KMS Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le politiche chiave o creare concessioni possono controllare l'accesso alle KMS chiavi se sono autorizzati a gestire tag o alias.
> Potrebbero essere necessari fino a cinque minuti prima che le modifiche ai tag e agli alias influiscano sull'autorizzazione delle chiavi. KMS Le modifiche recenti potrebbero essere visibili nelle API operazioni prima che influiscano sull'autorizzazione.
> Per controllare l'accesso a una KMS chiave in base al relativo alias, è necessario utilizzare una chiave di condizione. Non è possibile utilizzare un alias per rappresentare una KMS chiave nell'`Resource`elemento di una dichiarazione politica. Quando un alias appare

nell'`Resource`elemento, l'informativa sulla politica si applica all'alias, non alla chiave associata. KMS

Ulteriori informazioni

- Per dettagli sul AWS KMS supporto perABAC, inclusi esempi, vedere Utilizzate gli alias per controllare l'accesso alle chiavi KMS e. Utilizzate i tag per controllare l'accesso alle KMS chiavi
- Per informazioni più generali sull'uso dei tag per controllare l'accesso alle AWS risorse, vedi A cosa ABAC serve AWS? e Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida IAM per l'utente.

ABACchiavi di condizione per AWS KMS

Per autorizzare l'accesso alle KMS chiavi in base ai relativi tag e alias, utilizzate le seguenti chiavi condizionali in una politica o IAM in una politica chiave.

ABACchiave di condizione	Descrizione	Tipo di policy	AWS KMS operazioni
leggi: ResourceTag	Il tag (chiave e valore) sulla KMS chiave corrisponde al tag (chiave e valore) o al modello di tag nella politica	IAMsolo politica	KMSoperazioni relative alle risorse chiave [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	Politiche e IAM politiche chiave 1	TagResource, UntagResource
Leggi: TagKeys	Le chiavi tag nella richiesta corrispon	Politiche e IAM politiche chiave [1]	TagResource, UntagResource

ABACchiave di condizione	Descrizione	Tipo di policy	AWS KMS operazioni
	dono alle chiavi tag nella policy		
km: ResourceAliases	Gli alias associati alla KMS chiave corrispon dono agli alias o ai modelli di alias nella policy	IAMsolo politica	KMSoperazioni relative alle risorse chiave [2]
km: RequestAlias	L'alias che rappresen ta la KMS chiave nella richiesta corrisponde all'alias o ai modelli di alias nella politica.	Politiche e politiche chiave 1 IAM	Operazioni crittogra fiche, DescribeK eyGetPublicKey

[1] Qualsiasi chiave di condizione che può essere utilizzata in una politica chiave può essere utilizzata anche in una IAM politica, ma solo se la politica chiave lo consente.

[2] Un'operazione su una risorsa KMS chiave è un'operazione autorizzata per una KMS chiave particolare. Per identificare le operazioni KMS chiave relative alle risorse, nella tabella AWS KMS delle autorizzazioni, cerca un valore di KMS chiave nella `Resources` colonna relativa all'operazione.

Ad esempio, è possibile utilizzare queste chiavi di condizione per creare le seguenti policy.

- Una IAM politica `kms:ResourceAliases` che consente l'autorizzazione a utilizzare KMS chiavi con un alias o un pattern di alias particolare. Questa è leggermente diversa dalle politiche che si basano sui tag: sebbene sia possibile utilizzare modelli di alias in una politica, ogni alias deve essere unico in una regione and. Account AWS Ciò consente di applicare una politica a un set selezionato di KMS chiavi senza elencare la chiave ARNs delle KMS chiavi nell'informativa sulla politica. Per aggiungere o rimuovere KMS chiavi dal set, modificate l'alias della KMS chiave.
- Una politica chiave `kms:RequestAlias` che consente ai principali di utilizzare una KMS chiave in un'`Encrypt`operazione, ma solo quando la `Encrypt` richiesta utilizza quell'alias per identificare la chiave. KMS
- Una IAM politica `aws:ResourceTag/`*`tag-key`* che nega l'autorizzazione all'uso di KMS chiavi con una particolare chiave di tag e un valore di tag. Ciò consente di applicare una politica a un set

selezionato di KMS chiavi senza elencare la chiave ARNs delle KMS chiavi nell'informativa sulla politica. Per aggiungere o rimuovere KMS chiavi dal set, tagga o rimuovi il tag dalla KMS chiave.

- Una IAM politica `aws:RequestTag/`*`tag-key`* che consente ai responsabili di eliminare solo i tag `"Purpose"="Test"` KMS chiave.
- Una IAM politica `aws:TagKeys` che neghi il permesso di etichettare o rimuovere un tag da una KMS chiave con una `Restricted` chiave di tag.

ABACrende la gestione degli accessi flessibile e scalabile. Ad esempio, è possibile utilizzare la chiave `aws:ResourceTag/`*`tag-key`* condition per creare una IAM policy che consenta ai responsabili di utilizzare una KMS chiave per operazioni specifiche solo quando la KMS chiave ha un `Purpose=Test` tag. La politica si applica a tutte le KMS chiavi in tutte le regioni di. Account AWS

Se associata a un utente o a un ruolo, la seguente IAM politica consente ai responsabili di utilizzare tutte le KMS chiavi esistenti con un `Purpose=Test` tag per le operazioni specificate. Per fornire questo accesso a KMS chiavi nuove o esistenti, non è necessario modificare la politica. Basta attaccare il `Purpose=Test` tag alle KMS chiavi. Allo stesso modo, per rimuovere questo accesso dalle KMS chiavi con un `Purpose=Test` tag, modifica o elimina il tag.

```
{
  "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 consentire o negare inavvertitamente l'accesso a una chiave. KMS Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le politiche chiave o creare concessioni possono controllare l'accesso alle KMS chiavi se sono autorizzati a 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 Utilizzate gli alias per controllare l'accesso alle chiavi KMS e Utilizzate i tag per controllare l'accesso alle KMS chiavi.

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 AWS utilizzo particolari. 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 sei vicino alla quota di alias per KMS chiave, 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 chiave di tag per controllare l'accesso a più tipi di risorse, come un cluster Amazon Relational Database Service (RDSAmazon), un volume Amazon Elastic Block Store (EBSAmazon) e KMS una chiave. Questa funzione consente diversi modelli di autorizzazione più flessibili rispetto ai tradizionali controlli di accesso basati sui ruoli.
- Autorizza l'accesso a un gruppo di chiavi. KMS

  Puoi utilizzare i tag per gestire l'accesso a un gruppo di KMS chiavi nella stessa Account AWS regione. Assegna lo stesso tag o chiave di tag alle KMS chiavi che scegli. Quindi crea una semplice dichiarazione easy-to-maintain politica basata sul tag o sulla chiave del tag. Per aggiungere o rimuovere una KMS chiave dal tuo gruppo di autorizzazione, aggiungi o rimuovi il tag; non è necessario modificare la politica.

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, tra cui 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 nella richiesta. KMS Ad esempio, puoi concedere a un'autorizzazione principale l'uso di una KMS chiave in un'`Encrypt`operazione, ma solo quando il valore del `KeyId` parametro è. `alias/restricted-key-1` Per soddisfare questa condizione, è necessario quanto segue:

  - La KMS chiave deve essere associata a quell'alias.
  - La richiesta deve utilizzare l'alias per identificare la KMS chiave.
  - Il principale deve avere il permesso di utilizzare la KMS chiave in base alla `kms:RequestAlias` condizione.

  Ciò è particolarmente utile se le applicazioni utilizzano comunemente nomi alias o alias ARNs per fare riferimento alle KMS chiavi.
- Fornisci autorizzazioni molto limitate.

  Un alias deve essere univoco in una Account AWS regione and. Di conseguenza, concedere ai principali l'accesso a una KMS chiave basata su un alias può essere molto più restrittivo rispetto a concedere loro l'accesso in base a un tag. A differenza degli alias, i tag possono essere assegnati a più KMS chiavi nello stesso account e nella stessa regione. Se lo desideri, puoi utilizzare uno schema di alias, ad esempio per consentire agli `alias/test*` amministratori di accedere a un gruppo di KMS chiavi nello stesso account e nella stessa regione. Tuttavia, consentire o negare l'accesso a un particolare alias consente un controllo molto rigoroso sulle chiavi. KMS

Risoluzione dei problemi per ABAC AWS KMS

Il controllo dell'accesso alle KMS chiavi in base ai relativi tag e 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 suo valore viene modificato, ai mandanti che hanno accesso a una KMS chiave basata solo su quel tag verrà negato l'accesso alla chiave. KMS Ciò può verificarsi anche quando un tag incluso in una dichiarazione di rifiuto viene aggiunto a una KMS chiave. L'aggiunta di

un tag relativo alla politica a una KMS chiave può consentire l'accesso ai responsabili a cui dovrebbe essere negato l'accesso a una chiave. KMS

Ad esempio, supponiamo che un principale abbia accesso a una KMS chiave basata sul `Project=Alpha` tag, ad esempio l'autorizzazione fornita dalla seguente dichiarazione politica di esempio. IAM

```
{
  "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 da quella KMS chiave o il valore del tag viene modificato, il principale non è più autorizzato a utilizzare la KMS chiave 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, esamina CloudTrail i log e UntagResource le voci. TagResource

Per ripristinare l'accesso senza aggiornare la politica, modifica i tag sulla KMS chiave. 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 politiche per rilevare l'accesso che dipende dal tag e ottieni KMS le chiavi in tutte le regioni che hanno il 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 KMS chiave diversa, ai mandanti che hanno accesso alla KMS chiave solo in base a quell'alias verrà negato l'accesso alla chiave. KMS Ciò può verificarsi anche quando un alias associato a una KMS chiave è incluso in una dichiarazione di politica di negazione. L'aggiunta di un alias relativo alla politica a una KMS chiave può inoltre consentire l'accesso ai responsabili a cui dovrebbe essere negato l'accesso a una chiave. KMS

Ad esempio, la seguente dichiarazione di IAM politica utilizza la chiave kms: ResourceAliases condition per consentire l'accesso alle KMS chiavi in diverse regioni dell'account con uno qualsiasi degli alias specificati.

```
{
  "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 CreateAliasimmissioni. UpdateAliasDeleteAlias

Per ripristinare l'accesso senza aggiornare la politica, modifica l'alias associato alla chiave. KMS Poiché ogni alias può essere associato a una sola KMS chiave 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 alcuni principali su una KMS chiave può impedire allo stesso o ad altri principali di accedere a una chiave diversa. KMS

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 politiche per rilevare l'accesso che dipende dall'alias e trova le KMS chiavi in tutte le regioni associate all'alias.

Accesso negato a causa di quota alias

Gli utenti autorizzati a utilizzare una KMS chiave in base a una ResourceAliases condizione kms: riceveranno un'`AccessDenied`eccezione se la KMS chiave supera gli alias predefiniti per quota di KMS chiavi per quell'account e quella regione.

Per ripristinare l'accesso, elimina gli alias associati alla KMS chiave in modo che rispetti la quota. Oppure utilizzate un meccanismo alternativo per consentire agli utenti di accedere alla chiave. KMS

Modifica dell'autorizzazione ritardata

Le modifiche apportate ai tag e agli alias potrebbero richiedere fino a cinque minuti per influire sull'autorizzazione delle KMS chiavi. Di conseguenza, una modifica del tag o dell'alias potrebbe riflettersi nelle risposte delle API operazioni prima che influiscano sull'autorizzazione. È probabile che questo ritardo sia più lungo del breve ritardo di coerenza eventuale che influisce sulla maggior parte delle AWS KMS operazioni.

Ad esempio, potresti avere una IAM politica che consente a determinati principali di utilizzare qualsiasi KMS chiave con un `"Purpose"="Test"` tag. Quindi aggiungi il `"Purpose"="Test"` tag a una KMS chiave. Sebbene l'TagResourceoperazione sia stata completata e la ListResourceTagsrisposta confermi che il tag è stato assegnato alla KMS chiave, i responsabili potrebbero non avere accesso alla KMS chiave 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 si aggiorna un alias, si associa un alias esistente a una chiave diversa. KMS

La decrittografia e ReEncryptle richieste che specificano il nome o l'alias dell'alias ARN potrebbero non riuscire perché l'alias è ora associato a una KMS chiave che non ha crittografato il testo cifrato. Questa situazione in genere restituisce `IncorrectKeyException` o `NotFoundException`.

Oppure, se la richiesta ha il `DestinationKeyId` parametro no `KeyId` or, l'operazione potrebbe fallire con un'`AccessDenied`eccezione perché il chiamante non ha più accesso alla chiave che ha crittografato il testo cifrato. KMS

È possibile tracciare la modifica esaminando CloudTrail i log e le voci di CreateAliasregistro. UpdateAliasDeleteAlias È inoltre possibile utilizzare il valore del `LastUpdatedDate` campo nella ListAliasesrisposta per rilevare una modifica.

Ad esempio, la seguente risposta di ListAliasesesempio mostra che l'`ProjectAlpha_Test`alias nella `kms:ResourceAliases` condizione è stato aggiornato. Di conseguenza, i principali che hanno accesso in base all'alias perdono l'accesso alla chiave precedentemente associata. KMS Hanno invece accesso alla nuova chiave associataKMS.

```
$ 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 KMS chiave originale. Tuttavia, prima di agire, è necessario considerare l'effetto di tale modifica sulla KMS chiave attualmente associata. Se i principali utilizzassero quest'ultima KMS chiave nelle operazioni crittografiche, potrebbe essere necessario un accesso continuo ad essa. In questo caso, potresti voler aggiornare la politica per garantire che i principali abbiano il permesso di 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 KMS le chiavi 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.

RBACper 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 consente RBAC di controllare l'accesso alle chiavi specificando autorizzazioni granulari sull'utilizzo delle chiavi all'interno delle politiche chiave. Le politiche chiave specificano una risorsa, un'azione, un effetto, una condizione principale e facoltativa per concedere l'accesso alle chiavi.

Per implementarlo RBAC 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 `ExampleUserRole` IAM ruolo 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 IAM principali (ruoli o utenti) devono accedere alla chiave?
- Quali azioni deve eseguire ogni preside 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 `ExampleAdminRole` IAM ruolo di amministrare la KMS chiave.

```
{
            "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 necessarie ai 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

- IAMidentità (utenti, gruppi di utenti e ruoli)
- Tipi di controllo degli accessi

Consentire agli utenti di altri account di utilizzare una KMS chiave

Puoi consentire a utenti o ruoli diversi Account AWS di utilizzare una KMS chiave nel tuo account. L'accesso da più account richiede l'autorizzazione nella politica chiave della KMS chiave e in una IAM politica dell'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 responsabile di un altro account kms: ListKeys permission in una IAM policy o kms: ScheduleKeyDeletion permission on a KMS key in una policy chiave, i tentativi dell'utente di richiamare tali operazioni sulle tue risorse continueranno a fallire.

Per informazioni dettagliate sull'utilizzo KMS delle chiavi in diversi account per AWS KMS le operazioni, consulta la colonna Utilizzo tra account nella cartella and. AWS KMS autorizzazioni Utilizzo KMS delle chiavi in altri account C'è anche una sezione sull'uso tra account in ogni API descrizione della AWS Key Management Service API Guida di riferimento.

> **Warning**
>
> Fai attenzione nel concedere ai responsabili le autorizzazioni per usare le tue chiavi. KMS Quando possibile, segui il principio del privilegio minimo. Offri agli utenti l'accesso solo alle KMS chiavi di cui hanno bisogno solo per le operazioni necessarie.
> Inoltre, fai attenzione a non utilizzare una KMS chiave sconosciuta, specialmente una KMS chiave in un altro account. Gli utenti malintenzionati potrebbero concederti le autorizzazioni necessarie per utilizzare la loro KMS chiave per 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 autorizzare l'uso di una KMS chiave a utenti e ruoli in un altro account, devi utilizzare due diversi tipi di politiche:

- La politica chiave per la KMS chiave deve concedere all'account esterno (o agli utenti e ai ruoli nell'account esterno) l'autorizzazione a utilizzare la KMS chiave. La politica chiave si trova nell'account che possiede la KMS chiave.
- IAMle politiche dell'account esterno devono delegare le autorizzazioni relative alle policy chiave ai relativi utenti e ruoli. Queste policy sono impostate nell'account esterno e concedono autorizzazioni agli utenti e ai ruoli in tale account.

La politica chiave determina chi può avere accesso alla KMS chiave. La IAM politica determina chi ha accesso alla KMS chiave. Né la politica chiave né la IAM politica da sola sono sufficienti: è necessario modificarle entrambe.

Per modificare la politica chiave, è possibile utilizzare la Policy View in AWS Management Console o utilizzare le CreateKeyoperazioni or. PutKeyPolicy

Per informazioni sulla modifica delle IAM politiche, consultaUtilizzo IAM delle politiche con AWS KMS.

Per un esempio che mostra come la politica e IAM le politiche chiave interagiscono per consentire l'uso di una KMS chiave in un account diverso, vediEsempio 2: L'utente assume un ruolo con il permesso di utilizzare una KMS chiave in un'altra Account AWS.

Puoi visualizzare le AWS KMS operazioni risultanti su più account sulla KMS chiave nei tuoi AWS CloudTrail registri. Le operazioni che utilizzano KMS chiavi in altri account vengono registrate sia nell'account del chiamante che nell'account del proprietario della chiave. KMS

Argomenti

> **Note**
>
> Gli esempi in questo argomento mostrano come utilizzare congiuntamente una politica chiave e una IAM politica per fornire e limitare l'accesso a una KMS chiave. Questi esempi generici non intendono rappresentare le autorizzazioni richieste da un particolare Servizio AWS utente su una KMS chiave. Per informazioni sulle autorizzazioni richieste da an Servizio AWS , consultate l'argomento relativo alla crittografia nella documentazione del servizio.

Fase 1: aggiungere una dichiarazione di policy delle chiavi nell'account locale

La politica chiave di una KMS chiave è il fattore principale che determina chi può accedere alla KMS chiave e quali operazioni può eseguire. La politica chiave si trova sempre nell'account che possiede la KMS chiave. A differenza delle IAM politiche, le politiche chiave non specificano una risorsa. La risorsa è la KMS chiave associata alla politica chiave. Quando si fornisce l'autorizzazione per più account, la politica chiave relativa alla KMS chiave deve consentire all'account esterno (o agli utenti e ai ruoli nell'account esterno) l'autorizzazione a utilizzare la KMS chiave.

Per concedere a un account esterno l'autorizzazione a utilizzare la KMS chiave, aggiungi una dichiarazione alla politica chiave che specifichi l'account esterno. Nell'`Principal`elemento della policy chiave, inserisci l'Amazon Resource Name (ARN) dell'account esterno.

Quando specifichi un account esterno in una policy chiave, IAM gli amministratori dell'account esterno possono utilizzare IAM le policy 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 ai relativi responsabili sono effettive solo se l'account esterno è abilitato nella regione che ospita la chiave e la relativa politica KMS 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 di voler consentire all'account di utilizzare una chiave 444455556666 di crittografia simmetrica nell'account. KMS 111122223333 A tale scopo, aggiungete una dichiarazione politica come quella nell'esempio seguente alla politica chiave per la KMS chiave in account. 111122223333 Questa informativa sulla politica fornisce all'account esterno l'autorizzazione a utilizzare la KMS chiave nelle operazioni crittografiche per le chiavi di crittografia simmetriche. 444455556666 KMS

> **Note**
>
> L'esempio seguente rappresenta un esempio di policy chiave per la condivisione di una KMS chiave con un altro account. Sostituisci `Sid` l'`Principal`esempio e `Action` i valori con valori validi per l'uso previsto della tua KMS chiave.

```
{
    "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 KMS chiave finché IAM gli amministratori dell'account esterno non associano le IAM politiche appropriate alle proprie

identità. Le IAM politiche possono concedere l'autorizzazione a tutti o a un sottoinsieme degli utenti e dei ruoli esterni specificati nella politica chiave. Inoltre, possono consentire tutte o un sottoinsieme delle operazioni specificate nella policy delle chiavi.

La specificazione delle identità in una politica chiave limita le autorizzazioni che IAM gli amministratori dell'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 politica chiave dell'account che possiede la KMS chiave e creare IAM politiche nell'account dell'identità.

Per specificare utenti o ruoli esterni particolari in una policy chiave, nell'`Principal`elemento, inserisci l'Amazon Resource Name (ARN) di un utente o ruolo nell'account esterno.

Ad esempio, l'esempio seguente di informativa chiave consente `ExampleRole` a un account `444455556666` di utilizzare una KMS chiave nell'account`111122223333`. Questa dichiarazione sulla politica chiave fornisce all'account esterno il permesso di utilizzare la KMS chiave nelle operazioni crittografiche per le chiavi di crittografia simmetriche. `444455556666` KMS

> **Note**
>
> L'esempio seguente rappresenta un esempio di policy chiave per la condivisione di una KMS chiave con un altro account. Sostituisci `Sid` l'`Principal`esempio e `Action` i valori con valori validi per l'uso previsto della tua KMS chiave.

```
{
    "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 KMS chiave, a meno che un'altra dichiarazione politica non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua KMS chiave 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 il permesso di decrittografare ma non crittografare, oppure il permesso di visualizzare la KMS chiave ma non usarla. Per un elenco delle autorizzazioni sulle KMS chiavi, consulta. AWS KMS autorizzazioni

Impostazione della politica delle chiavi quando si crea una KMS chiave

Quando si utilizza l'CreateKeyoperazione per creare una KMS chiave, è possibile utilizzare il relativo `Policy` parametro per specificare una politica chiave che conceda a un account esterno, o a utenti e ruoli esterni, il permesso di utilizzare la KMS chiave.

Quando si crea una KMS chiave in AWS Management Console, si crea anche la relativa politica chiave. Quando si selezionano le identità nelle sezioni Amministratori chiave e Utenti chiave, AWS KMS aggiunge le istruzioni relative a tali identità alla politica KMS chiave della chiave. 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 istruzioni alla politica chiave. Questa operazione influisce solo sulla policy delle chiavi. Gli utenti e i ruoli dell'account esterno non possono utilizzare la KMS chiave finché non vengono allegate IAM politiche per concedere loro alcune o tutte queste autorizzazioni.

La prima dichiarazione politica chiave concede all'account esterno l'autorizzazione a utilizzare la KMS chiave nelle operazioni crittografiche. La seconda dichiarazione politica chiave consente all'account esterno di creare, visualizzare e revocare le concessioni sulla KMS chiave, 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 KMS chiavi che crittografano i dati degli utenti nei servizi AWS

Passaggio 2: aggiungere IAM le politiche nell'account esterno

La politica chiave dell'account proprietario della KMS chiave imposta l'intervallo valido per le autorizzazioni. Tuttavia, gli utenti e i ruoli dell'account esterno non possono utilizzare la KMS chiave finché non vengono allegate IAM politiche che delegano tali autorizzazioni o non si utilizzano concessioni per gestire l'accesso alla chiave. KMS Le IAM politiche sono impostate nell'account esterno.

Se la politica chiave fornisce l'autorizzazione all'account esterno, puoi allegare IAM le politiche a qualsiasi utente o ruolo dell'account. Tuttavia, se la politica chiave concede l'autorizzazione a utenti o ruoli specifici, la IAM politica può concedere tali autorizzazioni solo a tutti o a un sottoinsieme degli utenti e dei ruoli specificati. Se una IAM politica fornisce l'accesso KMS chiave ad altri utenti o ruoli esterni, non ha alcun effetto.

La politica chiave limita anche le azioni previste dalla IAM politica. La IAM politica può delegare tutte o un sottoinsieme delle azioni specificate nella politica chiave. Se la IAM politica elenca azioni che non sono specificate nella politica chiave, tali autorizzazioni non sono valide.

La seguente IAM politica di esempio consente al principale di utilizzare la KMS chiave in account `111122223333` per le operazioni crittografiche. 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 IAM policy per la condivisione di una KMS chiave con un altro account. Sostituite `Sid` l'`Resource`esempio e `Action` i valori con valori validi per l'uso previsto della KMS chiave.

```
{
    "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 politiche chiave, IAM le dichiarazioni politiche non contengono l'`Principal`elemento. Nelle IAM politiche, il principio è l'identità a cui è associata la politica.
- L'`Resource`elemento della IAM politica identifica la KMS chiave che il principale può utilizzare. Per specificare una KMS chiave, aggiungete la relativa chiave ARN all'`Resource`elemento.
- È possibile specificare più di una KMS chiave nell'`Resource`elemento. Ma se non specificate KMS chiavi particolari nell'`Resource`elemento, potreste inavvertitamente dare accesso a più KMS chiavi di quante ne intendiate.
- Per consentire all'utente esterno di utilizzare la KMS chiave con AWS i servizi che si integrano con AWS KMS, potrebbe essere necessario aggiungere autorizzazioni alla politica o alla politica chiave. IAM Per informazioni dettagliate, consultare Consentire l'uso di KMS chiavi esterne con Servizi AWS.

Per ulteriori informazioni sull'utilizzo delle IAM politiche, vederePolicy IAM.

Consentire l'uso di KMS chiavi esterne con Servizi AWS

Puoi concedere a un utente di un altro account il permesso di utilizzare la tua KMS chiave con un servizio integrato con AWS KMS. Ad esempio, un utente di un account esterno può utilizzare la tua KMS chiave per crittografare gli oggetti in un bucket Amazon S3 o per crittografare i segreti in cui sono archiviati. AWS Secrets Manager

La politica della chiave deve concedere all'utente esterno o all'account dell'utente esterno l'autorizzazione a utilizzare la chiave. KMS Inoltre, è necessario allegare delle IAM politiche all'identità che autorizza l'utente a utilizzare il Servizio AWS. Il servizio potrebbe inoltre richiedere che gli utenti dispongano di autorizzazioni aggiuntive nella politica o IAM nella politica chiave. 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 KMS delle chiavi in altri account

Se si dispone dell'autorizzazione per utilizzare una KMS chiave in un altro Account AWS, è possibile utilizzare la KMS chiave in AWS Management Console AWS SDKs, AWS CLI, e AWS Tools for PowerShell.

Per identificare una KMS chiave in un altro account in un comando o in una API richiesta di shell, utilizzate i seguenti identificatori di chiave.

- Per le operazioni crittografiche, e DescribeKeyGetPublicKey, utilizzate la chiave ARN o l'alias ARN della chiave. KMS
- Per CreateGrant, GetKeyRotationStatus, e ListGrantsRevokeGrant, usa la chiave ARN della KMS chiave.

Se inserisci solo un ID chiave o un nome alias, AWS si presuppone che la KMS chiave sia presente nel tuo account.

La AWS KMS console non visualizza KMS le chiavi degli altri account, anche se sei autorizzato a usarle. Inoltre, gli elenchi di KMS chiavi visualizzati nelle console di altri AWS servizi non includono KMS le chiavi di altri account.

Per specificare una KMS chiave in un altro account nella console di un AWS servizio, è necessario immettere la chiave ARN o l'alias ARN della KMS chiave. L'identificatore di chiave richiesto varia a seconda del servizio e potrebbe differire tra la console di servizio e le relative API operazioni. 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**
>
> Le dichiarazioni IAM politiche esistenti con caratteri jolly nel `Resource` campo ora si applicano sia alle chiavi a regione singola che a quelle multiregionali. Per limitarle alle chiavi a regione singola o alle KMS 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 politiche, IAM politiche e sovvenzioni chiave per consentire ai IAM committenti di gestire e utilizzare chiavi multiregionali nel tuo. Account AWS Ogni chiave multiregionale è una risorsa indipendente con una chiave e una politica chiave ARN uniche. È necessario stabilire e mantenere una politica chiave per ogni chiave e assicurarsi che IAM le politiche nuove ed esistenti implementino la strategia di autorizzazione.

Per supportare le chiavi multiregionali, AWS KMS utilizza un ruolo collegato al IAM servizio. 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

Nella progettazione di politiche e IAM politiche chiave per chiavi multiregionali, tenete conto dei seguenti principi.

- Politica chiave: ogni chiave multiregionale è una risorsa KMS chiave indipendente con una propria politica chiave. 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 CreateKeyand, 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.

- IAMcriteri: come per tutte le KMS chiavi, è possibile utilizzare i IAM criteri per controllare l'accesso alle chiavi multiregionali solo quando il criterio chiave lo consente. IAMi criteri 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 responsabili devono disporre `kms:CreateKey` dell'autorizzazione in una IAM politica applicabile alla regione in cui viene creata la chiave.

- Sovvenzioni: le sovvenzioni sono AWS KMS regionali. Ogni concessione consente le 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 su più KMS chiavi, anche se si tratta di chiavi multiregionali correlate.

- ChiaveARN: ogni chiave multiregionale ha una chiave unica. ARN 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 una dichiarazione IAM politica a una particolare chiave multiregionale, utilizza la relativa chiave ARN o uno ARN schema chiave che includa la regione. Per applicare una dichiarazione IAM politica a tutte le chiavi multiregionali correlate, utilizzate un carattere jolly (*) nell'elemento Region diARN, come illustrato nell'esempio seguente.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Describe*",
    "kms:List*"
  ],
  "Resource": {
      "arn:aws:kms:*::111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
  }
}
```

  Per applicare una dichiarazione di policy a tutte le chiavi multiregionali presenti nel sistema Account AWS, è possibile utilizzare la condizione kms: MultiRegion policy o un modello di ID chiave che includa 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, assume un ruolo collegato al servizio. AWS KMS 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 IAM politica che sia 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 IAM politica consente a un preside di creare qualsiasi tipo di KMS chiave.

```
{
  "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: MultiRegion condition. I valori validi sono `true` (Chiave multi-regione) o `false` (Chiave di singola regione). Ad esempio, la seguente dichiarazione IAM politica utilizza un'`Deny`azione con la chiave `kms:MultiRegion` condition per impedire ai principali di creare chiavi multiregionali.

```
{
  "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 IAM politica che è efficace nella regione della chiave di replica.

Presta attenzione quando consenti queste autorizzazioni. Consentono ai mandanti di creare KMS chiavi e le politiche chiave che ne autorizzano l'uso. 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'UpdatePrimaryRegionautorizzazione kms: in entrambe le regioni. Puoi fornire queste autorizzazioni in una politica o IAM in una politica chiave.

- `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 politica chiave offre agli utenti che possono assumere il ruolo di amministratore l'autorizzazione ad aggiornare la regione principale della KMS chiave. Questa KMS chiave 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 Regioni AWS capacità di ospitare una chiave primaria, usa la chiave kms: PrimaryRegion condition. Ad esempio, la seguente dichiarazione IAM politica consente ai principali di aggiornare la regione principale delle chiavi multiregione nella Account AWS, ma solo quando la nuova regione principale 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 responsabili che dispongono dell'autorizzazione a utilizzare e gestire KMS le chiavi in una regione Account AWS e hanno anche l'autorizzazione a utilizzare e gestire le chiavi multiregionali. 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 KMS chiave esistente, come Encrypt o. EnableKey

La seguente dichiarazione IAM politica di esempio utilizza la chiave `kms:MultiRegion` condition per impedire ai principali di utilizzare o gestire qualsiasi chiave multiregionale.

```
{
  "Effect": "Deny",
```

```
  "Action": "kms:*",
  "Resource": "*",
  "Condition": {
    "Bool": "kms:MultiRegion": true
  }
}
```

Questa dichiarazione IAM politica di esempio utilizza la `kms:MultiRegionKeyType` condizione per consentire ai responsabili di pianificare e annullare l'eliminazione delle chiavi, ma solo su chiavi di replica multiregionali.

```
{
  "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 KMS chiave, tutte le concessioni che si applicano alla KMS chiave e potenzialmente tutte le AWS Identity and Access Management (IAM) politiche. È possibile eseguire questa operazione per determinare l'ambito del potenziale utilizzo di una KMS chiave o per soddisfare i requisiti di conformità o di controllo. I seguenti argomenti possono aiutarti a generare un elenco completo dei AWS principali (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 politiche chiave sono il modo principale per controllare l'accesso alle KMS chiavi. Ogni KMS chiave ha esattamente una politica chiave.

Quando una politica chiave è costituita o include la politica chiave predefinita, la politica chiave consente IAM agli amministratori dell'account IAM di utilizzare le politiche per controllare l'accesso alla KMS chiave. Inoltre, se la politica chiave fornisce un'altra Account AWS autorizzazione all'uso della KMS chiave, IAM gli amministratori dell'account esterno possono utilizzare IAM le politiche per delegare tali autorizzazioni. Per determinare l'elenco completo dei responsabili che possono accedere alla KMS chiave, esamina le politiche. IAM

Per visualizzare la politica chiave di una chiave gestita AWS KMS dal cliente o Chiave gestita da AWSnel tuo account, utilizza l'GetKeyPolicyoperazione AWS Management Console o in. AWS KMS API Per visualizzare la politica chiave, è necessario disporre `kms:GetKeyPolicy` delle autorizzazioni per la KMS chiave. Per istruzioni sulla visualizzazione della politica chiave per una KMS chiave, vederethe 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 IAM utenti, IAM i ruoli e Account AWS l'`Principal`elemento hanno accesso a questa KMS chiave.

> **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 KMS chiave, a meno che un'altra dichiarazione politica non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua KMS chiave 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 un intestatario 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 di policy chiave quando si crea una nuova KMS chiave con o si crea una nuova KMS chiave a livello di codice ma non si fornisce una politica chiave. AWS Management Console

Un documento di policy chiave con una dichiarazione che consente l'accesso alle IAMpolitiche di abilitazione presenti nell'account per consentire l'accesso alla KMS chiave. Account AWS Ciò significa che gli utenti e i ruoli dell'account potrebbero avere accesso alla KMS chiave anche se non sono esplicitamente elencati come responsabili nel documento di policy chiave. Assicurati di esaminare tutte le IAM politiche 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` si riferisce al IAM ruolo indicato KMSKeyAdmins in Account AWS 111122223333. Gli utenti autorizzati ad assumere questo ruolo sono autorizzati a eseguire le azioni elencate nella dichiarazione politica, ovvero le azioni amministrative per la gestione di una KMS chiave.

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` si riferisce al IAM ruolo denominato EncryptionApp in Account AWS 111122223333. I responsabili autorizzati ad assumere questo ruolo sono autorizzati a eseguire le azioni elencate nella dichiarazione politica, che includono le operazioni crittografiche per una chiave di crittografia simmetrica. KMS

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` si riferisce al IAM ruolo denominato in 111122223333. EncryptionApp Account AWS I principali autorizzati ad assumere questo ruolo sono autorizzati a eseguire le operazioni elencate nell'istruzione della policy. Queste azioni, se combinate con le azioni consentite nell'esempio della dichiarazione politica 3, sono quelle necessarie per delegare l'uso della KMS chiave alla maggior parte dei servizi che si integrano con AWS KMS, in particolare ai AWS servizi che utilizzano sovvenzioni. 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 di autorizzazione.

Per scoprire tutti i diversi modi in cui è possibile specificare un principale in un documento di policy chiave, consulta Specificare un principale nella Guida per l'utente. IAM

Per ulteriori informazioni sulle politiche AWS KMS chiave, consultaPolitiche chiave in AWS KMS.

Analisi delle policy IAM

Oltre alla politica chiave e alle sovvenzioni, puoi anche utilizzare IAMle politiche per consentire l'accesso a una KMS chiave. Per ulteriori informazioni su come IAM le politiche e le politiche chiave interagiscono, consultaRisoluzione dei problemi relativi alle AWS KMS autorizzazioni.

Per determinare quali responsabili hanno attualmente accesso a una KMS chiave tramite IAM le policy, puoi utilizzare lo strumento IAMPolicy Simulator basato su browser oppure puoi effettuare richieste a. IAM API

Modi per esaminare le policy IAM

- Esame delle politiche con il simulatore di politiche IAM IAM
- Esaminando le politiche IAM con il IAM API

Esame delle politiche con il simulatore di politiche IAM IAM

Il IAM Policy Simulator può aiutarti a scoprire quali mandanti hanno accesso a una KMS chiave attraverso una politica. IAM

Utilizzare il simulatore di IAM policy per determinare l'accesso a una chiave KMS

1. Accedi a, quindi apri AWS Management Console il 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. (Facoltativo) Il Policy Simulator simula l'accesso a tutte le KMS chiavi per impostazione predefinita. Per simulare l'accesso a una KMS chiave specifica, scegli Impostazioni di simulazione, quindi digita l'Amazon Resource Name (ARN) della KMS chiave 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.

Esaminando le politiche IAM con il IAM API

È possibile utilizzare il IAM API per esaminare le IAM politiche a livello di codice. Le seguenti fasi forniscono una panoramica generale su come eseguire questa operazione:

1. Per ognuna delle entità Account AWS elencate come principali nella policy chiave (ovvero, ciascuna entitàAWS dell'account specificata in questo formato:`"Principal": {"AWS": "arn:aws:iam::111122223333:root"}`), utilizza le ListRolesoperazioni ListUsersand in IAM API per visualizzare tutti gli utenti e i ruoli nell'account.
2. Per ogni utente e ruolo nell'elenco, utilizzate l'SimulatePrincipalPolicyoperazione in IAMAPI, inserendo i seguenti parametri:
   - Per`PolicySourceArn`, specifica l'Amazon Resource Name (ARN) di un utente o di 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'`ActionNames`elenco, specifica ogni AWS KMS API azione da simulare. Per simulare tutte le AWS KMS API azioni, usa. `kms:*` Per testare AWS KMS API le singole azioni, fai precedere ogni API azione da "`kms:`«, ad esempio"»`kms:ListKeys`. Per un elenco completo delle AWS KMS API azioni, consulta Azioni nel AWS Key Management Service APIriferimento.
   - (Facoltativo) Per determinare se gli utenti o i ruoli hanno accesso a KMS chiavi specifiche, utilizza il `ResourceArns` parametro per specificare un elenco di Amazon Resource Names

(ARNs) delle KMS chiavi. Per determinare se gli utenti o i ruoli hanno accesso a qualsiasi KMS chiave, ometti il `ResourceArns` parametro.

IAMrisponde a ogni `SimulatePrincipalPolicy` richiesta con una decisione di valutazione:`allowed`,`explicitDeny`, o. `implicitDeny` Per ogni risposta che contiene una decisione di valutazione di`allowed`, la risposta include il nome dell' AWS KMS APIoperazione specifica consentita. Include anche ARN l'eventuale KMS chiave utilizzata 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 una KMS chiave può essere utilizzata. Le sovvenzioni sono allegate a una KMS chiave e ogni concessione contiene il principale che riceve l'autorizzazione a utilizzare la KMS chiave 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 KMS chiave, usa l' AWS KMS ListGrantsoperazione. Puoi esaminare le sovvenzioni alla ricerca di una KMS chiave per determinare chi o cosa ha attualmente accesso all'uso della KMS chiave tramite tali sovvenzioni. Ad esempio, quanto segue è una JSON rappresentazione di una concessione ottenuta dal comando list-grants in. 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 ha accesso all'uso della KMS chiave, cercate l'elemento. `"GranteePrincipal"` Nell'esempio precedente, il beneficiario principale è un utente con ruolo assunto associato all'istanza i-5d476fab. EC2 L'EC2infrastruttura utilizza questo ruolo per collegare

il volume crittografato vol-5cccfb4e all'istanza. EBS In questo caso, il ruolo di EC2 infrastruttura è autorizzato a utilizzare la KMS chiave perché in precedenza è stato creato un EBS volume crittografato protetto da questa chiave. KMS Il volume è stato quindi collegato a un'EC2istanza.

Di seguito è riportato un altro esempio di JSON rappresentazione di una concessione ottenuta dal comando list-grants in. AWS CLI Nel seguente esempio, il beneficiario principale 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 crittografica con una chiave asimmetrica o una KMS chiave. HMAC KMS Gli algoritmi e gli algoritmi asimmetrici non supportano un contesto di crittografiaMAC.

Tutte le operazioni AWS KMS crittografiche con KMSchiavi di crittografia simmetriche accettano un contesto di crittografia, un insieme 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. AWS KMS 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'CreateSnapshotoperazione Amazon Elastic Block Store EBS (Amazon), 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 AWS KMS keys tuo account. È possibile utilizzare il contesto di crittografia come vincolo nelle concessioni e come una condizione nelle istruzioni di policy.

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 AWS Key Management Service e EncryptionContext sul blog sulla AWS sicurezza.

Ulteriori informazioni sul contesto di crittografia.

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 politiche o nelle IAM politiche chiave, non sarà possibile specificare il contesto di crittografia nelle chiavi delle condizioni delle politiche, 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 i dettagli sulle regole IAM dei documenti politici, consulta i requisiti relativi ai IAM nomi nella Guida IAM per l'utente.

Contesto di crittografia nelle policy

Il contesto di crittografia viene utilizzato principalmente per verificare l'integrità e l'autenticità. Ma è anche possibile utilizzare il contesto di crittografia per controllare l'accesso alla crittografia simmetrica nelle politiche e AWS KMS keys IAM nelle politiche chiave.

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 dichiarazione di politica chiave consente al `RoleForExampleApp` ruolo di utilizzare la chiave nelle operazioni. KMS `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 si crea una concessione, è possibile 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 KMS chiavi e dati è stato effettuato l'accesso. La voce di registro mostra esattamente quali KMS chiavi sono state utilizzate per crittografare o decrittografare dati specifici a cui fa riferimento il contesto di crittografia nella voce di registro.

> **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?

`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 AWS KMS API chiamate con AWS CloudTrail.

Specificazione DryRun con l'API

Per utilizzarloDryRun, 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'CreateGrantoperazione 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
```

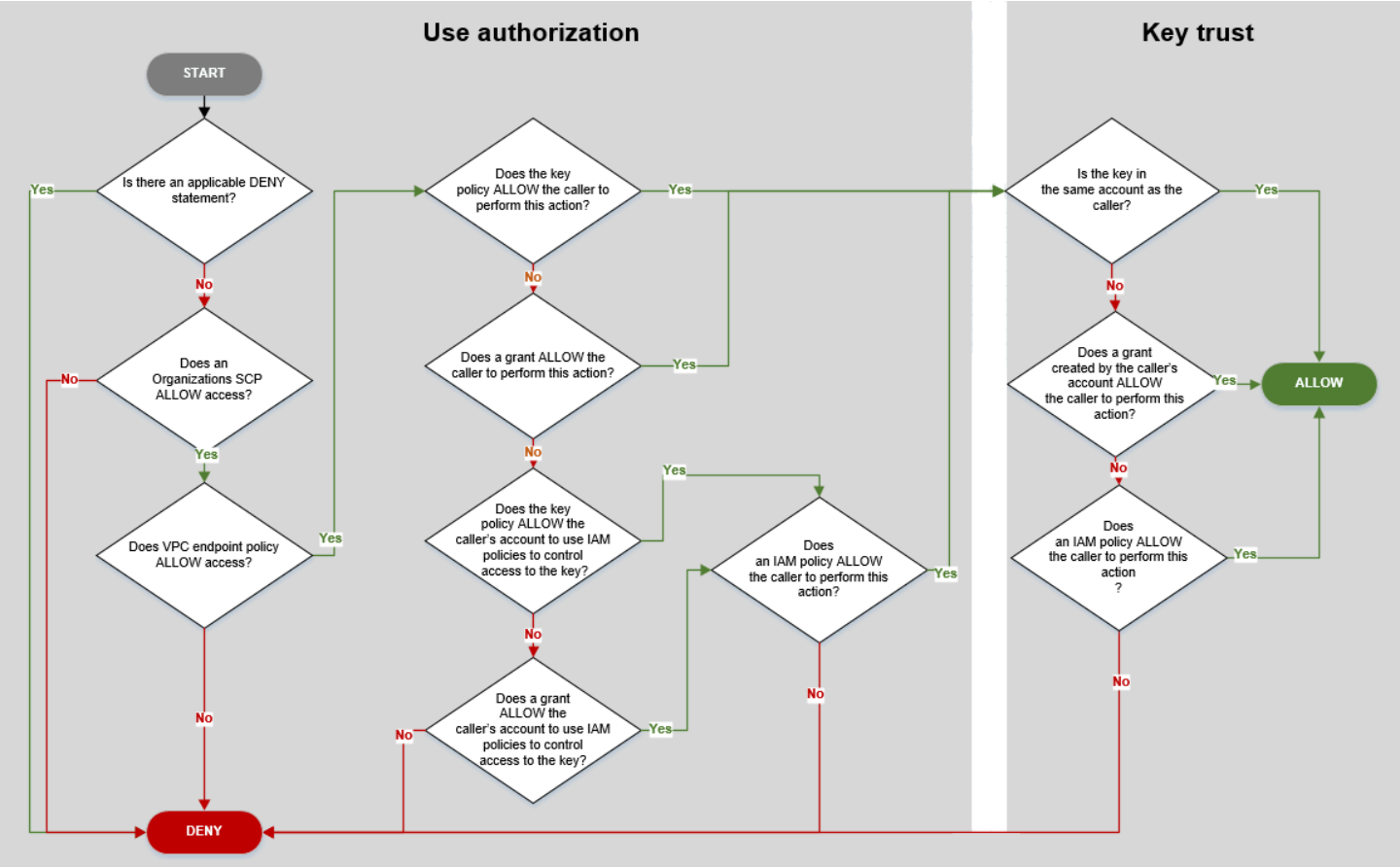
Risoluzione dei problemi relativi alle AWS KMS autorizzazioni

Quando si autorizza l'accesso a una KMS chiave, AWS KMS valuta quanto segue:

- La politica chiave allegata alla KMS chiave. La politica chiave è sempre definita nella regione Account AWS e nella regione che possiede la KMS chiave.
- Tutte IAMle politiche allegate all'utente o al ruolo che effettua la richiesta. IAMle politiche che regolano l'uso di una KMS chiave da parte di un principale sono sempre definite nella chiave principale. Account AWS
- Tutte le sovvenzioni che si applicano alla KMS chiave.
- Altri tipi di politiche che potrebbero essere applicate alla richiesta di utilizzo della KMS chiave, come le politiche di controllo dei AWS Organizations servizi e le politiche degli VPC endpoint.

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 di policy per determinare se l'accesso alla KMS chiave 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 all'uso determina se è consentito utilizzare una KMS chiave in base alla politica chiave, alle IAM politiche, alle concessioni e ad altre politiche applicabili.
- La fiducia nelle chiavi determina se è necessario considerare attendibile una KMS chiave che è consentito utilizzare. In generale, ti fidi delle risorse di cui disponi Account AWS. Tuttavia, puoi sentirti sicuro di utilizzare KMS le chiavi anche in un altro modo Account AWS se una sovvenzione o una IAM politica del tuo account ti consentono di utilizzare la KMS chiave.

Puoi utilizzare questo diagramma di flusso per scoprire perché a un chiamante è stata concessa o negata l'autorizzazione a utilizzare una KMS chiave. È anche possibile utilizzarlo per valutare le policy e le autorizzazioni. Ad esempio, il diagramma di flusso mostra che a un chiamante può essere negato l'accesso mediante una `DENY` dichiarazione esplicita o in assenza di una `ALLOW` dichiarazione esplicita nella politica, nella politica o nella concessione della chiave. IAM

Il diagramma di flusso è in grado di spiegare alcuni scenari comuni.

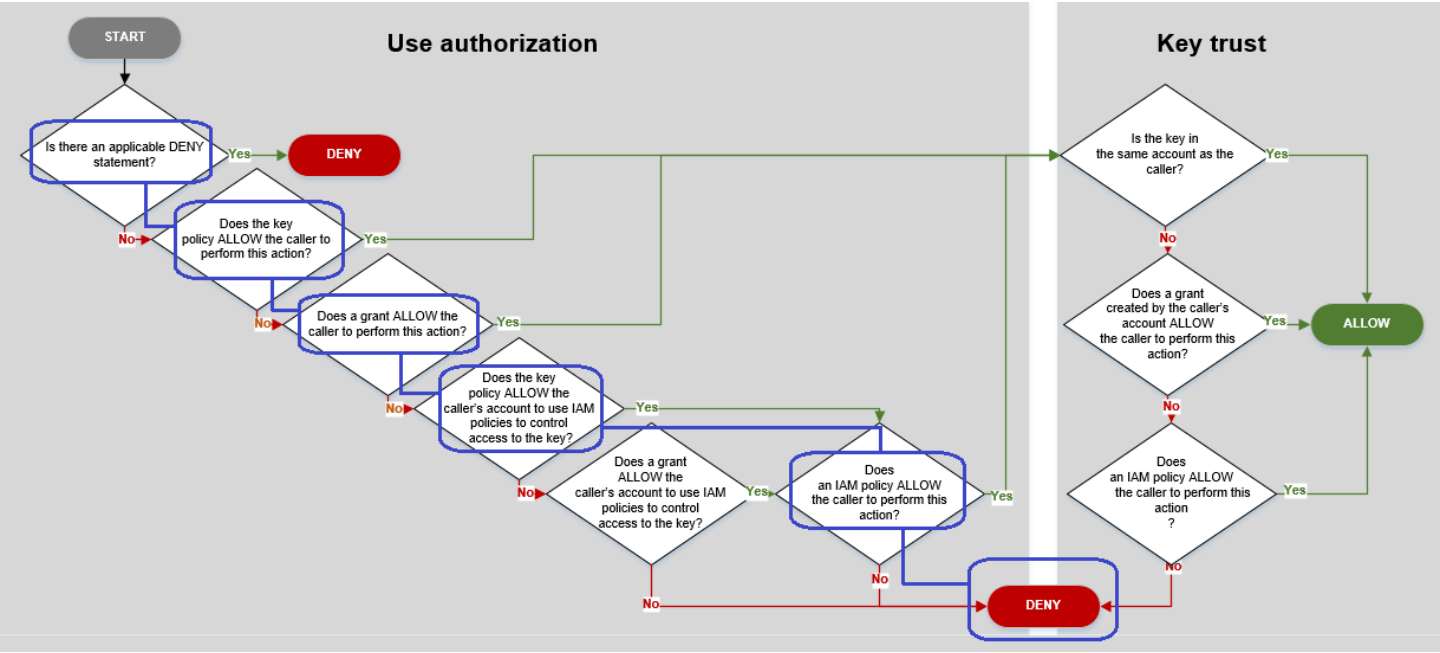
Esempi di autorizzazione

- Esempio 1: all'utente viene negato l'accesso a una chiave nel proprio KMS Account AWS
- Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una KMS chiave in un'altra Account AWS

Esempio 1: all'utente viene negato l'accesso a una chiave nel proprio KMS Account AWS

Alice è un IAM utente del 111122223333 Account AWS. Le è stato negato l'accesso a una KMS chiave dello stesso. Account AWS Perché Alice non può usare la KMS chiave?

In questo caso, ad Alice viene negato l'accesso alla KMS chiave perché non esiste una politica, una IAM politica o una concessione chiave che le dia le autorizzazioni richieste. La politica chiave della KMS chiave consente di utilizzare IAM le politiche Account AWS per controllare l'accesso alla KMS chiave, ma nessuna IAM politica dà ad Alice il permesso di usare la KMS chiave.



Considerare le relative policy per questo esempio.

- La KMS chiave che Alice vuole usare ha la politica delle chiavi predefinita. Questa politica consente a chi possiede Account AWS la KMS chiave IAM di utilizzare le politiche per controllare l'accesso alla KMS chiave. Questa politica chiave soddisfa la politica Does the key account ALLOW the callers to use IAM policy per controllare l'accesso alla chiave? condizione nel diagramma di flusso.

```
{
  "Version" : "2012-10-17",
  "Id" : "key-test-1",
  "Statement" : [ {
    "Sid" : "Delegate to IAM policies",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

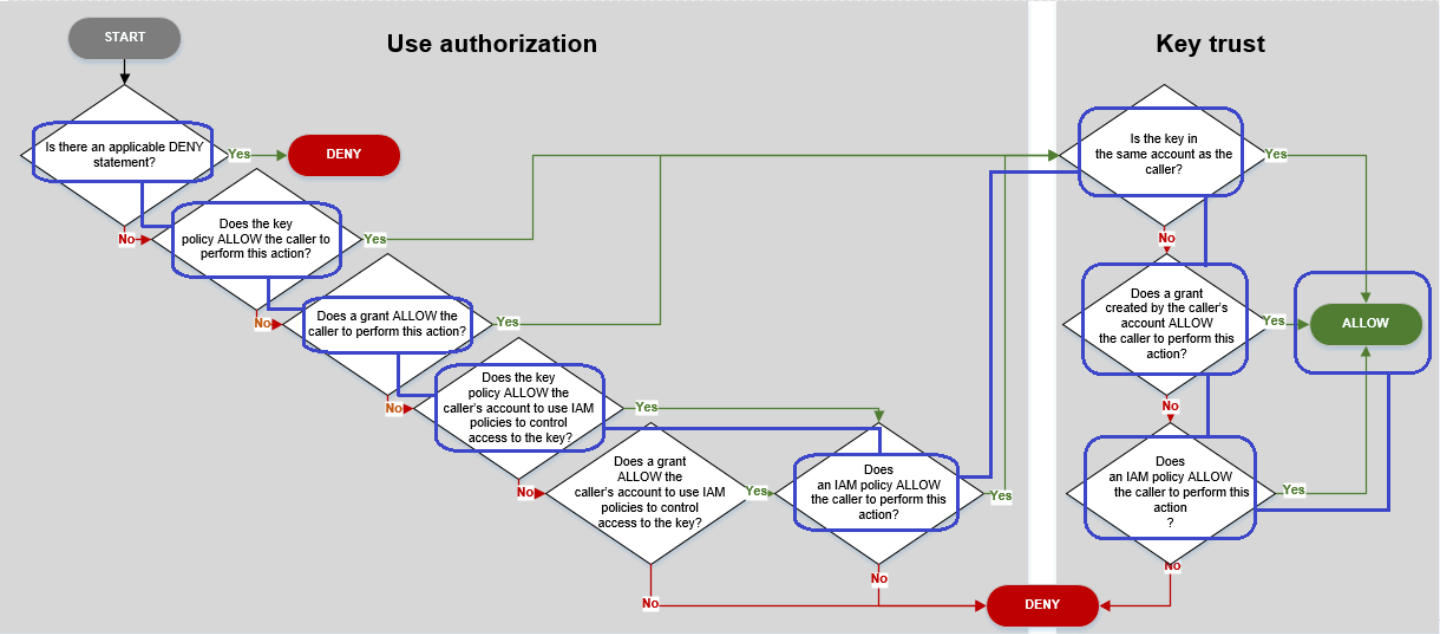
- Tuttavia, nessuna politica, IAM politica o concessione chiave dà ad Alice il permesso di usare la KMS chiave. Pertanto, ad Alice viene negato il permesso di usare la KMS chiave.

Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una KMS chiave in un'altra Account AWS

Bob è un utente nell'account 1 (111122223333). Gli è consentito utilizzare una KMS chiave nell'account 2 (444455556666) nelle operazioni crittografiche. In che modo è possibile?

> **Tip**
>
> Quando valuti le autorizzazioni per più account, ricorda che la politica della chiave è specificata nell'account della chiave. KMS La IAM politica è specificata nell'account del chiamante, anche quando il chiamante utilizza un account diverso. Per informazioni dettagliate su come fornire l'accesso alle KMS chiavi su più account, consulta. Consentire agli utenti di altri account di utilizzare una KMS chiave

- La politica chiave per la KMS chiave nell'account 2 consente all'account 2 IAM di utilizzare le politiche per controllare l'accesso alla KMS chiave.
- La politica chiave per la KMS chiave nell'account 2 consente all'account 1 di utilizzare la KMS chiave nelle operazioni crittografiche. Tuttavia, l'account 1 deve utilizzare IAM politiche per consentire ai suoi principali di accedere alla KMS chiave.
- Una IAM policy nell'account 1 consente al `Engineering` ruolo di utilizzare la KMS chiave nell'account 2 per operazioni crittografiche.
- Bob, un utente nell'account 1, ha l'autorizzazione per assumere il ruolo `Engineering`.
- Bob può fidarsi di questa KMS chiave, perché anche se non è presente nel suo account, una IAM politica del suo account gli dà il permesso esplicito di usare questa KMS chiave.



Considerate le politiche che consentono a Bob, un utente dell'account 1, di utilizzare la KMS chiave nell'account 2.

- La politica chiave per la KMS chiave consente all'account 2 (444455556666, l'account che possiede la KMS chiave) di utilizzare IAM le politiche per controllare l'accesso alla chiave. KMS Questa politica chiave consente inoltre all'account 1 (111122223333) di utilizzare la KMS chiave nelle operazioni crittografiche (specificate nell'`Action`elemento dell'informativa). Tuttavia, nessuno nell'account 1 può utilizzare la KMS chiave nell'account 2 finché l'account 1 non definisce IAM le politiche che consentono ai principali di accedere alla chiave. KMS

Nel diagramma di flusso, questa politica chiave dell'account 2 soddisfa la politica chiave utilizzata IAM dall'account ALLOW del chiamante per controllare l'accesso alla chiave? condizione.

```
{
    "Id": "key-policy-acct-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to use IAM policies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow account 1 to use this KMS key",
            "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 IAM politica del chiamante Account AWS (account 1, 111122223333) fornisce l'autorizzazione principale a eseguire operazioni crittografiche utilizzando la KMS chiave 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 IAM politiche relative a più account come questa sono efficaci solo quando la politica chiave per la chiave dell'account 2 autorizza l'account 1 a utilizzare la KMS 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, ciò soddisfa la IAMpolitica Does an a the caller di eseguire questa azione? condizione.

```
{
    "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": "AROA4KJY2TU23Y7NK62MV"
    }
}
```

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, è necessario essere autorizzati a utilizzare una KMS chiave in un'operazione di crittografia.

Per controllare l'accesso alle AWS KMS risorse, utilizza le politiche, le politiche e IAMleconcessioni chiave. Ogni KMS chiave deve avere una politica chiave. Se la politica chiave lo consente, puoi anche utilizzare IAM politiche e sovvenzioni per consentire ai mandanti l'accesso alla KMS chiave. 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. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAMIdentity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando () per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta AWS Signature Version 4 per API le richieste nella Guida per l'IAMutente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'AWS IAM Identity Center utente e Autenticazione a AWS più fattori IAM nella Guida per l'IAMutente.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'IAMutente.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per informazioni su IAM Identity Center, vedi Cos'è IAM Identity Center? nella Guida AWS IAM Identity Center per l'utente.

IAM users and groups

Un IAMutente è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente. IAM

Un gruppo IAM è un'identità che specifica una raccolta di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile assegnare un nome a un gruppo IAMAdminse concedere a tale gruppo le autorizzazioni per amministrare le risorse. IAM

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per IAM gli utenti nella Guida per l'IAMutente.

Ruoli IAM

Un IAMruolo è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM ma non è associato a una persona specifica. Per assumere temporaneamente un IAM ruolo in AWS Management Console, puoi passare da un utente a un IAM ruolo (console). È possibile assumere un ruolo chiamando un' AWS APIoperazione AWS CLI or o utilizzando

un'operazione personalizzataURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Metodi per assumere un ruolo nella Guida per l'IAMutente.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, consulta Creare un ruolo per un provider di identità di terze parti (federazione) nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in. IAM Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni IAM utente temporanee: un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (principale attendibile) di un account diverso di accedere alle risorse nel tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse su più account IAM nella Guida per l'utente. IAM
- Accesso tra servizi: alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che quindi avvia un'altra azione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. FASle richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di

servizio da IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.

- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare, ma non modificare le autorizzazioni dei ruoli collegati ai servizi.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'EC2istanza e che effettuano AWS CLI o richiedono AWS API. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'EC2istanza e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida per l'IAMutente.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere Panoramica delle JSON politiche nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione

`iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, da o da AWS API. AWS CLI

Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente. IAM

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo. Account AWS Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta Scegliere tra politiche gestite e politiche in linea nella Guida per l'IAMutente.

Policy basate su risorse

Una politica AWS KMS chiave è una politica basata sulle risorse che controlla l'accesso a una chiave. KMS Ogni KMS chiave deve avere una politica chiave. È possibile utilizzare un altro meccanismo di autorizzazione per consentire l'accesso alla KMS chiave, ma solo se la politica della chiave lo consente. (È possibile utilizzare una IAM politica per negare l'accesso a una KMS chiave anche se la politica chiave non lo consente esplicitamente.)

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa, ad esempio una KMS chiave, per controllare l'accesso alla 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 politiche AWS gestite daIAM, ad esempio la politica `AWSKeyManagementServicePowerUser` gestita, in una politica basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo). IAM IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM
- Politiche di controllo del servizio (SCPs): SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCPLimita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations andSCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Criteri di controllo delle risorse (RCPs): RCPs sono JSON criteri che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le IAM politiche allegate a ciascuna risorsa di tua proprietà. RCPLimita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations eRCPs, incluso un elenco di Servizi AWS tale supportoRCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per sapere come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle politiche nella Guida per l'IAMutente.

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 KMS chiave. Alcune AWS KMS operazioni consentono di utilizzare un alias per identificare una KMS chiave.

Ogni istanza di una KMS chiave o 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 di ARN:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`Regione AWS`*`:`*`Account AWS ID`*`:key/`*`key ID`*

  Esempio:ARN

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

- Alias

  Formato di ARN:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`Regione AWS`*`:`*`Account AWS ID`*`:alias/`*`alias name`*

  Esempio:ARN

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

AWS KMS fornisce una serie di API operazioni per utilizzare le AWS KMS risorse. Per ulteriori informazioni sull'identificazione KMS delle chiavi nelle AWS KMS API operazioni AWS Management

Console and, vedereIdentificatori chiave () KeyId. Per un elenco delle AWS KMS operazioni, vedere il AWS Key Management Service APIriferimento.

Crea una chiave KMS.

È possibile creare AWS KMS keys in o utilizzando l' AWS Management ConsoleCreateKeyoperazione o la AWS CloudFormation risorsaAWS::KMS: :Key. Durante questo processo, impostate la politica chiave per la KMS chiave, che potete modificare in qualsiasi momento. Selezionate anche i seguenti valori che definiscono il tipo di KMS chiave che create. Non è possibile modificare queste proprietà dopo la creazione della KMS chiave.

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 in modalità Galois Counter Mode (GCM) per fornire crittografia/decrittografia autenticate di dati di dimensioni inferiori AES 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 conto dell'utente. Servizi AWS

- RSA, curva ellittica o (solo regioni SM2 della Cina) tasti asimmetrici

  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 relative ai codici di autenticazione dei messaggi () basate su hash HMAC

  Queste chiavi sono chiavi a 256 bit utilizzate per le operazioni di firma e verifica.

  KMSle chiavi 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 delle chiavi è una proprietà che determina le operazioni crittografiche supportate dalla chiave. KMSle chiavi possono utilizzare come chiave`ENCRYPT_DECRYPT`, `SIGN_VERIFYGENERATE_VERIFY_MAC`, o`KEY_AGREEMENT`. Ogni KMS chiave può avere un solo

utilizzo. L'utilizzo di una KMS chiave 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 KMS le chiavi, la specifica chiave determina se la KMS chiave è simmetrica o asimmetrica. Determina anche il tipo di materiale della chiave e gli algoritmi supportati.

La specifica chiave predefinita, SYMMETRIC_, rappresenta una chiave di crittografia simmetrica a 256 DEFAULT bit. Per una descrizione dettagliata di tutte le specifiche chiave supportate, vedere. Riferimento alle specifiche chiave

Origine dei materiali chiave

L'origine del materiale KMS chiave è una proprietà chiave che identifica l'origine del materiale chiave contenuto nella KMS chiave. L'origine del materiale chiave viene scelta quando si crea la KMS chiave e non è possibile modificarla. La fonte del materiale chiave influisce sulle caratteristiche di sicurezza, durabilità, disponibilità, latenza e velocità effettiva della KMS chiave.

Ogni KMS chiave include un riferimento al relativo materiale chiave nei metadati. L'origine del materiale chiave delle KMS chiavi di crittografia simmetriche può variare. È possibile utilizzare materiale chiave AWS KMS generato, materiale chiave generato in un archivio di chiavi personalizzato o importare materiale chiave personalizzato.

Per impostazione predefinita, ogni KMS chiave ha un materiale chiave unico. Tuttavia, è possibile creare un set di chiavi per più regioni con lo stesso materiale chiave.

KMSle chiavi possono avere uno dei seguenti valori chiave di origine del materiale:`AWS_KMS`, `EXTERNAL` (materiale chiave importato), `AWS_CLOUDHSM` (KMS AWS CloudHSM chiave in un archivio chiavi) o `EXTERNAL_KEY_STORE` (KMSchiave in un archivio chiavi esterno).

Argomenti

- Autorizzazioni per la creazione di chiavi KMS
- Scelta del tipo di KMS chiave da creare
- Crea una chiave di crittografia simmetrica KMS
- Creare una chiave asimmetrica KMS

- Crea una HMAC KMS chiave
- Creazione di chiavi primarie multiregionali
- Creazione di chiavi di replica multiregionali
- Crea una KMS chiave con materiale chiave importato
- Creare una KMS chiave in un archivio di AWS CloudHSM chiavi
- Creare una KMS chiave in archivi di chiavi esterni

Autorizzazioni per la creazione di chiavi KMS

Per creare una KMS chiave nella console o utilizzando laAPIs, è necessario disporre della seguente autorizzazione in una IAM politica. Quando possibile, utilizzare le chiavi di condizione per limitare le autorizzazioni. Ad esempio, è possibile utilizzare la chiave kms: KeySpec condition in una IAM policy per consentire ai principali di creare solo chiavi di crittografia simmetriche.

Per un esempio di IAM policy per i dirigenti che creano le chiavi, vedi. Consenti 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, consultare ABACper AWS KMS.

- kms: CreateKey è obbligatorio.
- kms: CreateAlias è necessario per creare una KMS chiave 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, consultare Controlla l'accesso alle chiavi multiregionali.

Il kms: non è richiesta PutKeyPolicy l'autorizzazione per creare la KMS chiave. L'autorizzazione `kms:CreateKey` include l'autorizzazione per impostare la policy chiave iniziale. Tuttavia, è necessario aggiungere questa autorizzazione alla politica chiave durante la creazione della KMS

chiave per assicurarsi di poter controllare l'accesso alla KMS chiave. L'alternativa è utilizzare il BypassLockoutSafetyCheckparametro, che non è consigliato.

KMSle chiavi appartengono all' AWS account in cui sono state create. L'IAMutente che crea una KMS chiave non è considerato il proprietario della chiave e non dispone automaticamente dell'autorizzazione per utilizzare o gestire la KMS chiave che ha creato. Come qualsiasi altro responsabile, il creatore della chiave deve ottenere l'autorizzazione tramite una politica, una IAM politica o una concessione chiave. 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 KMS chiave da creare

Il tipo di KMS chiave che si crea dipende in gran parte dal modo in cui si prevede di utilizzare la KMS chiave, dai requisiti di sicurezza e dai requisiti di autorizzazione. Il tipo di chiave e l'utilizzo della KMS chiave determinano le operazioni crittografiche che la chiave può eseguire. Ogni KMS chiave può essere utilizzata solo una chiave. L'utilizzo di una KMS chiave per più di un tipo di operazione rende il prodotto di tutte le operazioni più vulnerabile agli attacchi.

Per consentire ai principali di creare KMS chiavi solo per un particolare utilizzo della chiave, usa la chiave kms: KeyUsage condition. Puoi anche usare la chiave `kms:KeyUsage` condition per consentire ai principali di richiamare API le operazioni per una KMS chiave in base al suo utilizzo. Ad esempio, è possibile consentire l'autorizzazione a disabilitare una KMS chiave solo se l'utilizzo della chiave è SIGN _VERIFY.

Utilizza le seguenti indicazioni per determinare il tipo di KMS chiave necessario in base al tuo caso d'uso.

Crittografia e decrittografia dei dati

Utilizza una KMSchiave simmetrica per la maggior parte dei casi d'uso che richiedono la crittografia e la decrittografia 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 KMS chiave richiede che sia il mittente che il destinatario dei dati crittografati dispongano di AWS credenziali valide per la chiamata. 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 KMSchiavi asimmetriche sono una buona scelta. È possibile distribuire la

chiave pubblica della chiave asimmetrica KMS per consentire a questi utenti di crittografare i dati. Inoltre, le applicazioni che devono decrittografare tali dati possono utilizzare la chiave privata della chiave asimmetrica interna. KMS AWS KMS

Firma dei messaggi e verifica delle firme

Per firmare messaggi e verificare le firme, è necessario utilizzare una chiave asimmetrica. KMS È possibile utilizzare una KMS chiave con una specifica chiave che rappresenti una coppia di RSA chiavi, una coppia di chiavi a curva ellittica () ECC o una coppia di SM2 chiavi (solo regioni della Cina). La specifica della chiave scelta è determinata dall'algoritmo di firma che desideri utilizzare. Gli algoritmi di ECDSA firma supportati dalle coppie di ECC chiavi sono consigliati rispetto agli algoritmi di firma. RSA Tuttavia, potrebbe essere necessario utilizzare una specifica chiave e un algoritmo di firma particolari 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, è necessario utilizzare una chiave KMS asimmetrica con una specifica chiave o RSASM2una specifica chiave (solo regioni della Cina). Per crittografare i dati AWS KMS con la chiave pubblica di una coppia di KMS chiavi, utilizzare l'operazione Encrypt. 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 asimmetricaKMS, puoi usarla all'esterno di. AWS KMS Ma non è più soggetto ai controlli di sicurezza che proteggono la KMS chiave in ingresso. 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'`Decrypt`operazione ha esito negativo se i dati sono stati crittografati con una chiave pubblica da una chiave che utilizza una KMS chiave di. `SIGN_VERIFY` L'operazione avrà esito negativo anche 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 GetPublicKeyrisposta forniscono le informazioni da includere quando si condivide la chiave pubblica.

Ricava segreti condivisi

Per ricavare segreti condivisi, usa una KMS chiave con curva ellittica NIST consigliata o materiale chiave (solo per le regioni SM2della Cina). AWS KMS utilizza il cofattore di crittografia a curva ellittica 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' DeriveSharedSecretoperazione 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 i NISTconsigli per la derivazione delle chiavi quando si utilizza il segreto condiviso non elaborato per derivare una chiave simmetrica.

Genera e verifica codici HMAC

Per generare e verificare codici di autenticazione dei messaggi basati su hash, utilizza una HMAC KMS chiave. Quando create una HMAC chiave in AWS KMS, AWS KMS crea e protegge il materiale chiave e garantisce l'utilizzo MAC degli algoritmi corretti per la chiave. HMACi codici possono essere utilizzati anche come numeri pseudo-casuali e in determinati scenari per la firma e la tokenizzazione simmetriche.

HMACKMSle chiavi sono chiavi simmetriche. Quando crei una HMAC KMS chiave nella AWS KMS console, scegli il tipo di `Symmetric` chiave.

Utilizza con AWS i servizi

Per creare una KMS chiave da utilizzare con un AWS servizio integrato AWS KMS, consulta la documentazione del servizio. AWS i servizi che crittografano i dati richiedono una chiave di crittografia simmetrica. KMS

Oltre a queste considerazioni, le operazioni crittografiche su KMS chiavi con specifiche chiave diverse hanno prezzi diversi e quote di richieste diverse. 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.

Crea una chiave di crittografia simmetrica KMS

Questo argomento spiega come creare la KMS chiave di base, una chiave di crittografia simmetrica per una singola regione con materiale KMS chiave proveniente da. AWS KMSÈ possibile utilizzare questa KMS chiave per proteggere le risorse in un. Servizio AWS

È possibile creare KMS chiavi di crittografia simmetriche nella AWS KMS console utilizzando o utilizzando CreateKeyAPIil modello AWS::KMS: AWS CloudFormation :Key.

La specifica chiave predefinita, SYMMETRIC_ DEFAULT, è la specifica chiave per le chiavi di crittografia simmetriche. KMS 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'CreateKeyoperazione, se non si specifica un valore, viene selezionato _. `KeySpec` SYMMETRIC DEFAULT Se non avete motivo di utilizzare una specifica chiave diversa, SYMMETRIC _ DEFAULT è una buona scelta.

Per informazioni sulle quote che si applicano alle KMS chiavi, consulta. Quote

Utilizzo della console AWS KMS

È possibile utilizzare AWS Management Console per creare AWS KMS keys (KMSchiavi).

> **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 AWS Management Console 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 di crittografia simmetrica, per Tipo di KMS chiave scegli Simmetrico.
6. In Utilizzo della chiave, l'opzione Crittografa e decrittografa è selezionata per impostazione predefinita.
7. Scegli Next (Successivo).
8. Digitate 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 ABACper AWS KMS e Utilizzate gli alias per controllare l'accesso alle chiavi KMS.

Un alias è un nome visualizzato che è possibile utilizzare per identificare la chiave. KMS Ti consigliamo di scegliere un alias che indichi il tipo di dati che intendi proteggere o l'applicazione che intendi utilizzare con la KMS chiave.

Gli alias sono necessari quando si crea una KMS chiave in. AWS Management Console Sono facoltativi quando si utilizza l'CreateKeyoperazione.

9. (Facoltativo) Digitate una descrizione per la KMS chiave.

   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 KMS chiave AWS Management Console o utilizza l'UpdateKeyDescriptionoperazione.

10. (Facoltativo) Digita tag per una chiave di un valore di tag facoltativo. Per aggiungere più di un tag alla KMS chiave, scegli Aggiungi tag.

    > **Note**
    >
    > L'aggiunta di tag o detag a una KMS chiave può consentire o negare l'autorizzazione alla chiave. KMS Per informazioni dettagliate, consulta ABACper AWS KMS e Utilizzate i tag per controllare l'accesso alle KMS 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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

11. Scegli Next (Successivo).

12. Seleziona gli IAM utenti e i ruoli che possono amministrare la KMS chiave.

> **Note**
>
> Questa politica chiave offre il Account AWS pieno controllo di questa KMS chiave. Consente agli amministratori degli account IAM di utilizzare le politiche per concedere ad altri responsabili il permesso di gestire la KMS chiave. Per informazioni dettagliate, consultare the section called “Policy delle chiavi predefinita”.
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> 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.

13. (Facoltativo) Per impedire agli IAM utenti e ai ruoli selezionati di eliminare questa KMS chiave, nella sezione Eliminazione della chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.
14. Scegli Next (Successivo).
15. Seleziona gli IAM utenti e i ruoli che possono utilizzare la chiave nelle operazioni crittografiche

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori delle istruzioni `"Allow use of the key"` 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 KMS chiave 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 responsabili degli account esterni di utilizzare la KMS chiave, gli amministratori dell'account esterno devono creare IAM politiche che forniscano tali autorizzazioni. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

17. Scegli Next (Successivo).
18. Rivedi le principali dichiarazioni politiche relative alla 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 Fine per creare la KMS chiave.

Usando il AWS KMS API

È possibile utilizzare l'CreateKeyoperazione 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 Utilizzare `CreateKey` con un AWS SDK o 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'GenerateDataKeyoperazione.

L'`CreateKey`operazione non consente di specificare un alias, ma è possibile utilizzare l'CreateAliasoperazione 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 di crittografia simmetrica con materiale KMS 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 si specifica una politica di chiave per la nuova KMS chiave, la politica di chiave predefinita `CreateKey` applicata è diversa dalla politica di chiave predefinita applicata dalla console quando la si utilizza per creare una nuova chiave. KMS

Ad esempio, questa chiamata all'GetKeyPolicyoperazione restituisce la politica chiave `CreateKey` applicabile. Fornisce l' Account AWS accesso alla KMS chiave e le consente di creare politiche AWS Identity and Access Management (IAM) per la KMS chiave. Per informazioni dettagliate sulle IAM politiche e sulle politiche chiave per le KMS chiavi, vedere KMSaccesso con chiavi e autorizzazioni

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

```
{
  "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" : "*"
  } ]
}
```

Creare una chiave asimmetrica KMS

È possibile creare KMSchiavi asimmetriche nella AWS KMS console, utilizzando o utilizzando il CreateKeyAPImodello::: :Key. AWS KMS AWS CloudFormation Una chiave asimmetrica rappresenta una coppia di KMS 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, vedereScarica la chiave pubblica.

Quando si crea una chiave asimmetrica, è necessario selezionare una specifica KMS 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, vedere. Riferimento alle specifiche chiave

AWS i servizi che si integrano con AWS KMS non supportano le chiavi asimmetricheKMS. Se desideri creare una KMS chiave che crittografa i dati archiviati o gestiti in un AWS servizio, crea una chiave di crittografia simmetrica. KMS

Per informazioni sulle autorizzazioni necessarie per creare KMS le chiavi, consulta. Autorizzazioni per la creazione di chiavi KMS

Utilizzo della console AWS KMS

È possibile utilizzare il AWS Management Console per creare AWS KMS keys (KMSchiavi) asimmetriche. Ogni chiave asimmetrica rappresenta una coppia di KMS chiavi pubblica e privata.

> **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 AWS Management Console 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 asimmetrica, in Tipo di KMS chiave, scegli Asimmetrico.
6. Per creare una chiave asimmetrica per la crittografia a chiave pubblica, in Utilizzo KMS delle chiavi, scegli Crittografa e decrittografa.

   Per creare una chiave asimmetrica per la firma dei messaggi e la verifica delle firme, in Utilizzo delle KMS chiavi, scegli Firma e verifica.

   Per creare una chiave asimmetrica per derivare segreti condivisi, in Utilizzo delle KMS chiavi, scegli Contratto chiave.

   Per informazioni sulla scelta di un valore di utilizzo chiave, consulta Scelta del tipo di KMS chiave da creare.
7. Seleziona una specifica (specifica chiave) per la tua chiave asimmetrica. KMS
8. Scegli Next (Successivo).
9. Digitate 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 utilizzare per identificare la KMS chiave nella console e in alcune altre. AWS KMS APIs Ti consigliamo di scegliere un alias che indichi il tipo di dati che intendi proteggere o l'applicazione che intendi utilizzare con la KMS chiave.

   Gli alias sono necessari quando si crea una KMS chiave in. AWS Management Console Non è possibile specificare un alias quando si utilizza l'CreateKeyoperazione, ma è possibile utilizzare

la console o l'CreateAliasoperazione per creare un alias per una chiave esistente. KMS Per informazioni dettagliate, consultare Alias in AWS KMS.

10. (Facoltativo) Digitare una descrizione per la KMS chiave.

    Inserisci una descrizione che spieghi il tipo di dati che intendi proteggere o l'applicazione che intendi utilizzare con la KMS chiave.

    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 KMS chiave utilizzata AWS Management Console o utilizza l'UpdateKeyDescriptionoperazione.

11. (Facoltativo) Digita tag per una chiave di un valore di tag facoltativo. Per aggiungere più di un tag alla KMS chiave, scegli 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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

12. Scegli Next (Successivo).

13. Seleziona gli IAM utenti e i ruoli che possono amministrare la KMS chiave.

> **Note**
>
> Questa politica chiave offre il Account AWS pieno controllo di questa KMS chiave. Consente agli amministratori degli account IAM di utilizzare le politiche per concedere ad altri responsabili il permesso di gestire la KMS chiave. Per informazioni dettagliate, consultare the section called “Policy delle chiavi predefinita”.
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> 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 agli IAM utenti e ai ruoli selezionati di eliminare questa KMS chiave, nella sezione Eliminazione della chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.
15. Scegli Next (Successivo).
16. Seleziona gli IAM utenti e i ruoli che possono utilizzare la KMS chiave per le operazioni crittografiche.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori delle istruzioni `"Allow use of the key"` 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 KMS chiave 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 responsabili degli account esterni di utilizzare la KMS chiave, gli amministratori dell'account esterno devono creare IAM politiche che forniscano tali autorizzazioni. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

18. Scegli Next (Successivo).
19. Rivedi 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 KMS chiave.

Usando il AWS KMS API

È possibile utilizzare l'CreateKeyoperazione 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 si crea una KMS chiave asimmetrica, è necessario specificare il `KeySpec` parametro che determina il tipo di chiavi create. Inoltre, è necessario specificare `KeyUsage` il valore ENCRYPT _DECRYPT, _ o SIGN _VERIFY. KEY AGREEMENT Non è possibile modificare queste proprietà dopo la creazione della KMS chiave.

L'`CreateKey`operazione non consente di specificare un alias, ma è possibile utilizzare l'CreateAliasoperazione per creare un alias per la nuova KMS chiave.

> **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 KMS chiavi asimmetrica per la crittografia pubblica

L'esempio seguente utilizza l'`CreateKey`operazione per creare una chiave asimmetrica di chiavi a 4096 RSA bit progettata per la crittografia a KMS 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 KMS key pair asimmetrica per la firma e la verifica

Il comando di esempio seguente crea una chiave asimmetrica che rappresenta una coppia di KMS chiavi utilizzate per la firma e la verificaECC. 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 KMS key pair asimmetrica per derivare segreti condivisi

Il comando di esempio seguente crea una chiave asimmetrica che rappresenta una coppia di KMS chiavi utilizzate per derivare segreti condivisi. ECDH 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
    }
}
```

Crea una HMAC KMS chiave

È possibile creare HMAC KMS chiavi nella AWS KMS console utilizzando il CreateKeyAPIo utilizzando il AWS CloudFormation modelloAWS::KMS: :Key.

Quando si crea una HMAC KMS chiave, è necessario selezionare una specifica chiave. AWS KMS supporta più specifiche chiave per HMAC KMS le chiavi. 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 necessarie per creare KMS le chiavi, vedereAutorizzazioni per la creazione di chiavi KMS.

Utilizzo della console AWS KMS

È possibile utilizzare il AWS Management Console per creare HMAC KMS chiavi. HMACKMSle chiavi sono chiavi simmetriche con un utilizzo chiave di Generate and verify. MAC È inoltre possibile creare chiavi HMAC multiregionali.

1. Accedi a AWS Management Console 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. Alla voce Key type (Tipo di chiave), scegliere Symmetric (Simmetrica).

   HMACKMSi tasti sono simmetrici. Utilizzi la stessa chiave per generare e verificare HMAC i tag.
6. Per Utilizzo della chiave, scegli Genera e verifica MAC.

   Genera e verifica MAC è l'unico utilizzo valido HMAC KMS delle chiavi.

   > **Note**
   >
   > L'utilizzo delle chiavi viene visualizzato per le chiavi simmetriche solo quando HMAC KMS le chiavi sono supportate nella regione selezionata.
7. Seleziona una specifica (specifica chiave) per la tua chiave. 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 sicure.
8. Per creare una HMAC chiave primaria multiregionale, in Opzioni avanzate, scegli Chiave multiregionale. Le proprietà condivise che definisci per questa KMS chiave, come il tipo di chiave e l'utilizzo della chiave, verranno condivise con le relative chiavi di replica.

   Non è possibile utilizzare questa procedura per creare una chiave di replica. Per creare una chiave di replica multiregionale, segui le istruzioni per creare una HMAC 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.

    Si consiglia di utilizzare un alias che identifichi la KMS chiave come HMAC chiave, ad esempio. `HMAC/test-key` In questo modo sarà più facile identificare HMAC le chiavi 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 della chiave.

    Gli alias sono necessari quando si crea una KMS chiave in. AWS Management Console Non è possibile specificare un alias quando si utilizza l'CreateKeyoperazione, ma è possibile utilizzare la console o l'CreateAliasoperazione per creare un alias per una chiave esistente. KMS Per informazioni dettagliate, consultare Alias in AWS KMS.

11. (Facoltativo) Immettere una descrizione per la KMS chiave.

    Inserisci una descrizione che spieghi il tipo di dati che intendi proteggere o l'applicazione che intendi utilizzare con la KMS chiave.

    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 KMS chiave AWS Management Console contenuta AWS Management Console o utilizza l'UpdateKeyDescriptionoperazione.

12. (Facoltativo) Inserisci un tag della chiave e un valore facoltativo. Per aggiungere più di un tag alla KMS chiave, scegli Aggiungi tag.

    Prendi in considerazione l'aggiunta di un tag che identifichi la chiave come HMAC chiave, ad esempio. `Type=HMAC` In questo modo sarà più facile identificare HMAC le chiavi 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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

13. Scegli Next (Successivo).

14. Seleziona gli IAM utenti e i ruoli che possono amministrare la KMS chiave.

> **Note**
>
> Questa politica chiave offre il Account AWS pieno controllo di questa KMS chiave. Consente agli amministratori degli account IAM di utilizzare le politiche per concedere ad altri responsabili il permesso di gestire la KMS chiave. Per informazioni dettagliate, consultare the section called “Policy delle chiavi predefinita”.
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> 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.

15. (Facoltativo) Per impedire agli IAM utenti e ai ruoli selezionati di eliminare questa KMS chiave, nella sezione Eliminazione della chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.
16. Scegli Next (Successivo).
17. Seleziona gli IAM utenti e i ruoli che possono utilizzare la KMS chiave per le operazioni crittografiche.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori delle istruzioni `"Allow use of the key"` 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 KMS chiave 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 responsabili degli account esterni di utilizzare la KMS chiave, gli amministratori dell'account esterno devono creare IAM politiche che forniscano tali autorizzazioni. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

19. Scegli Next (Successivo).
20. Rivedi 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 Fine per creare la HMAC KMS chiave.

Usando il AWS KMS API

È possibile utilizzare l'CreateKeyoperazione per creare una HMAC KMS chiave. Questi esempi utilizzano la AWS Command Line Interface (AWS CLI), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Quando si crea una HMAC KMS chiave, è necessario specificare il `KeySpec` parametro che determina il tipo di KMS chiave. Inoltre, è necessario specificare `KeyUsage` il valore GENERATE _ VERIFY _MAC, anche se è l'unico valore di utilizzo delle chiavi valido per HMAC le chiavi. Per creare una HMAC KMS chiave multiregionale, aggiungete il `MultiRegion` parametro con il valore di`true`. Non è possibile modificare queste proprietà dopo la creazione della KMS chiave.

L'`CreateKey`operazione non consente di specificare un alias, ma è possibile utilizzare l'CreateAliasoperazione per creare un alias per la nuova KMS chiave. Si consiglia di utilizzare un alias che identifichi la KMS chiave come chiave, ad esempioHMAC. `HMAC/test-key` In questo modo sarà più facile identificare HMAC le chiavi 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 HMAC KMS chiave in un ambiente Regione AWS in cui HMAC le chiavi non sono supportate, l'`CreateKey`operazione restituisce un `UnsupportedOperationException`

L'esempio seguente utilizza l'`CreateKey`operazione per creare una HMAC KMS chiave 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
    }
}
```

Creazione di chiavi primarie multiregionali

È possibile creare una chiave primaria multiregionale nella AWS KMS console o utilizzando. AWS KMS API 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 KMS chiave, inclusa l'CreateKeyautorizzazione kms: in una policy. IAM Il principale 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 chiave primaria multiregionale, considera attentamente IAM gli utenti e i ruoli che scegli per amministrare e utilizzare la chiave. IAMle politiche possono concedere ad altri IAM utenti e ruoli l'autorizzazione a gestire la KMS chiave.

IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

Utilizzo della AWS KMS console

Per creare una chiave primaria multiregionale nella AWS KMS console, utilizza lo stesso processo che utilizzeresti per creare qualsiasi KMS chiave. 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 AWS Management Console 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. Selezionare un tipo di chiave simmetrico o asimmetrico. Le chiavi simmetriche sono le chiavi di default.

   È possibile creare chiavi simmetriche e asimmetriche multiregione, incluse chiavi multiregione, che sono simmetriche. HMAC KMS
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 “Creare una chiave asimmetrica KMS” o the section called “Crea una HMAC KMS chiave”.
7. Espandere Advanced options (Opzioni avanzate).
8. In Origine del materiale chiave, per AWS KMS generare il materiale chiave che condivideranno le chiavi primarie 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 puoi 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. È possibile assegnare alla chiave primaria multiregionale e alle relative 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 ABACper AWS KMS e Utilizzate 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 multiregionale 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. È possibile modificare i tag sui KMS tasti in qualsiasi momento.

    > **Note**
    >
    > L'aggiunta di tag o detag a una KMS chiave può consentire o negare l'autorizzazione all'uso della chiave. KMS Per informazioni dettagliate, consulta ABACper AWS KMS e Utilizzate i tag per controllare l'accesso alle KMS chiavi.

13. Seleziona IAM gli utenti e i ruoli che possono amministrare 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 politica chiave di una KMS chiave in qualsiasi momento.
> - Quando crei una chiave primaria multiregionale, prendi in considerazione l'utilizzo della politica 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 agli IAM utenti e ai ruoli selezionati di eliminare questa KMS chiave, nella sezione Eliminazione della chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.
15. Scegli Next (Successivo).
16. Seleziona gli IAM utenti e i ruoli che possono utilizzare la KMS chiave per le operazioni crittografiche.

> **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.

17. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa KMS chiave 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 responsabili degli account esterni di utilizzare la KMS chiave, gli amministratori dell'account esterno devono creare IAM politiche che forniscano tali autorizzazioni. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

18. Scegli Next (Successivo).
19. Esamina 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.

Usando il AWS KMS API

Per creare una chiave primaria multiregionale, utilizzare l'CreateKeyoperazione. 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 multiregionali sono gli stessi valori predefiniti per tutte le altre KMS chiavi, inclusa la politica delle chiavi predefinita. Questa procedura crea una chiave di crittografia simmetrica, la chiave predefinita. KMS

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": [ ]
      }
    }
}
```

Creazione di chiavi di replica multiregionali

È possibile creare una chiave di replica multiarea nella AWS KMS console, utilizzando l'ReplicateKeyoperazione o utilizzando un modello AWS:::: KMS. ReplicaKey AWS CloudFormation Non è possibile utilizzare l'CreateKeyoperazione per creare una chiave di replica.

È possibile utilizzare queste procedure per replicare qualsiasi chiave primaria multiregionale, inclusa una chiave di crittografia simmetrica, una KMS chiave KMSasimmetrica o una chiave. HMAC KMS

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 multiregione 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, riprovate `KMSInvalidStateException` o richiamate DescribeKeyper 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 nella 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 IAM utenti e i ruoli che scegli per amministrare e utilizzare la chiave di replica. IAMle politiche possono concedere ad altri IAM utenti e ruoli l'autorizzazione a gestire la chiave. KMS
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

Utilizzo della AWS KMS console

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 KMS chiave singola regionale standard 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 AWS Management Console 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. Si apre la pagina dei dettagli chiave della 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 ABACper AWS KMS e Utilizzate 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 multiregionale 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'aggiunta di tag o detag a una KMS chiave può consentire o negare l'autorizzazione alla chiave. KMS Per informazioni dettagliate, consulta ABACper AWS KMS e Utilizzate i tag per controllare l'accesso alle KMS chiavi.

11. Seleziona IAM gli utenti e i ruoli che possono amministrare 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. Puoi modificare la politica chiave di qualsiasi KMS chiave in qualsiasi momento.

- La AWS KMS console aggiunge gli amministratori chiave alla politica 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.

12. (Facoltativo) Per impedire agli IAM utenti e ai ruoli selezionati di eliminare questa KMS chiave, nella sezione Eliminazione della chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.
13. Scegli Next (Successivo).
14. Seleziona gli IAM utenti e i ruoli che possono utilizzare la KMS chiave per le operazioni crittografiche.

> **Nota**
>
> 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.

15. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa KMS chiave 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 responsabili degli account esterni di utilizzare la KMS chiave, gli amministratori dell'account esterno devono creare IAM politiche che forniscano tali autorizzazioni. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

16. Scegli Next (Successivo).

17. Rivedi 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.

Usando il AWS KMS API

Per creare una chiave di replica multiregionale, utilizzare l'ReplicateKeyoperazione. Non è possibile utilizzare l'CreateKeyoperazione 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 KMS le chiavi, non i valori della chiave primaria.

> **Note**
>
> Se non specificate valori per i `Tags` parametri`Description`, o`KeyPolicy`, 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 la chiave ARN 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 KMS chiave con materiale chiave importato

Il materiale chiave importato consente di proteggere le AWS risorse con le chiavi crittografiche generate dall'utente. Il materiale chiave importato è associato a una KMS chiave particolare. È possibile reimportare lo stesso materiale chiave nella stessa KMS chiave, ma non è possibile importare materiale chiave diverso nella KMS chiave e non è possibile convertire una KMS chiave progettata per il materiale chiave importato in una KMS chiave con materiale AWS KMS chiave.

La seguente panoramica illustra come importare il materiale della chiave in AWS KMS. Per ulteriori dettagli su ogni fase del processo, consulta l'argomento corrispondente.

1. Create una KMS chiave senza materiale chiave: l'origine deve essereEXTERNAL. 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 KMS chiave. In una fase successiva importerai il tuo materiale chiave in questa KMS chiave.

   Il materiale chiave che importate deve essere compatibile con le specifiche chiave della chiave associata AWS KMS . Per ulteriori informazioni sulla compatibilità, consulta 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 elementi proteggono il tuo materiale della chiave mentre viene importato in AWS KMS.

   In questo passaggio, scegli il tipo («specifica chiave») della chiave di RSA wrapping e l'algoritmo di wrapping che utilizzerai per crittografare i dati in transito. 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. Decripta 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 diventa inutilizzabile. KMS Per continuare a utilizzare la KMS chiave, è necessario reimportare lo stesso materiale chiave.

   Quando l'operazione di importazione viene completata correttamente, lo stato della KMS chiave cambia da a`PendingImport`. `Enabled` È ora possibile utilizzare la KMS chiave nelle operazioni crittografiche.

AWS KMS registra una voce nel AWS CloudTrail registro quando si crea la KMS chiave, 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 KMS chiavi con materiale chiave importato, l'utente deve disporre dell'autorizzazione per le operazioni di questo processo. È possibile fornire le `kms:GetParametersForImport` autorizzazioni e `kms:DeleteImportedKeyMaterial` le autorizzazioni nella politica chiave al momento della creazione della KMS chiave. `kms:ImportKeyMaterial` 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 KMS chiavi con materiale chiave importato, il principale necessita delle seguenti autorizzazioni.

- kms: CreateKey (politica) IAM
  - Per limitare questa autorizzazione alle KMS chiavi con materiale chiave importato, usa 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 o IAM politica chiave)
  - 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 o politica chiave) 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 `KMSAdminRole` autorizzare l'esempio a gestire tutti gli aspetti di una KMS chiave con materiale chiave importato, includi una dichiarazione politica chiave come la seguente nella politica chiave della KMS chiave.

```
{
  "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 chiave che importate deve essere compatibile con le specifiche chiave della KMS chiave associata. Per le coppie di chiavi asimmetriche, importate solo la chiave privata della coppia. AWS KMS ricava la chiave pubblica dalla chiave privata.

AWS KMS supporta le seguenti specifiche chiave per KMS le chiavi con materiale chiave importato.

KMSspecifiche chiave	Requisiti del materiale della chiave
Chiavi di crittografia simmetrica SYMMETRIC_DEFAULT	256 bit (32 byte) di dati binari Nelle regioni cinesi, devono essere dati binari a 128 bit (16 byte).
HMACtasti HMAC_224 HMAC_256 HMAC_384 HMAC_512	HMACil materiale chiave deve essere conforme al 2104. RFC La lunghezza della chiave deve corrispondere alla lunghezza specificata dalle specifiche della chiave.
RSAchiave privata asimmetrica RSA_2048 RSA_3072 RSA_4096	La chiave privata RSA asimmetrica importata deve far parte di una coppia di chiavi conforme a 3447. RFC Modulo: 2048 bit, 3072 bit o 4096 bit Numero di numeri primi: 2 (le chiavi con più numeri primi non sono supportate) RSA Il materiale a chiave asimmetrica deve essere BER codificato o codificato nel formato Public-Key DER Cryptography Standards () #8 conforme alla norma 5208. PKCS RFC

KMSspecifiche chiave	Requisiti del materiale della chiave
Chiave privata asimmetrica a curva ellittica ECC_ NIST _P256 (secp256r1) ECCNIST_ P384 (secp384r1) ECCNIST_ P521 (secp521r1) ECC_ SECG _P256K1 (secp256k1)	La chiave privata ECC asimmetrica importata deve far parte di una coppia di chiavi conforme a 5915. RFC Curva: NIST P-256, P-384, NIST P-521 o SecP256k1 NIST Parametri: solo curve con nome (le chiavi con parametri espliciti vengono rifiutate) ECC Coordinate pubbliche dei punti: possono essere compresse, non compresse o proiettive Il materiale a chiave asimmetrica deve essere BER codificato o codificato nel formato Public-Key DER Cryptography Standards () #8 conforme alla norma 5208. PKCS RFC
SM2chiave privata asimmetrica (solo per le regioni cinesi)	La chiave privata SM2 asimmetrica importata deve far parte di una coppia di chiavi conforme a GM/T 0003. Curva: SM2 Parametri: solo curva con nome (SM2le chiavi con parametri espliciti vengono rifiutate) Coordinate pubbliche dei punti: possono essere compresse, non compresse o proiettive Il materiale a chiave asimmetrica deve essere BER codificato o codificato nel formato Public-Key DER Cryptography Standards () #8 conforme alla norma 5208. PKCS RFC

Fase 1: Creare un materiale AWS KMS key senza chiave

Per impostazione predefinita, AWS KMS crea automaticamente del materiale chiave quando si crea una KMS chiave. Per importare invece il tuo materiale chiave, inizia creando una KMS chiave senza materiale chiave. Quindi, importare il materiale chiave. Per creare una KMS chiave senza materiale chiave, usa la AWS KMS console o l'CreateKeyoperazione.

Per creare una chiave senza materiale della chiave, specifica un'origine `EXTERNAL`. La proprietà di origine di una KMS chiave è immutabile. Una volta creata, non è possibile convertire una KMS chiave progettata per il materiale chiave importato in una KMS chiave con materiale chiave proveniente AWS KMS o da qualsiasi altra fonte.

Lo stato chiave di una KMS chiave con un'`EXTERNAL`origine e senza materiale chiave è`PendingImport`. Una KMS chiave può rimanere invariata a `PendingImport` tempo indeterminato. Tuttavia, non è possibile utilizzare una KMS chiave in `PendingImport` stato nelle operazioni crittografiche. Quando importate materiale chiave, lo stato della KMS chiave cambia in ed è possibile utilizzarlo nelle operazioni crittografiche. `Enabled`

AWS KMS registra un evento nel AWS CloudTrail registro quando si crea la KMS chiave, 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 KMS chiave senza materiale chiave (console)
- Creazione di una KMS chiave senza materiale chiave ()AWS KMS API

Creazione di una KMS chiave senza materiale chiave (console)

È sufficiente creare una KMS chiave per il materiale chiave importato una sola volta. È possibile importare e reimportare lo stesso materiale chiave nella KMS chiave esistente tutte le volte che è necessario, ma non è possibile importare materiale chiave diverso in una KMS chiave. Per informazioni dettagliate, consultare Fase 2: download della chiave pubblica di wrapping e del token di importazione.

Per trovare KMS le chiavi esistenti con materiale chiave importato nella tabella delle chiavi gestite dai clienti, utilizza l'icona a forma di ingranaggio nell'angolo in alto a destra per mostrare la colonna

Origin nell'elenco delle KMS chiavi. Il valore di Origine per le chiavi importate è Esterna (Importa il materiale della chiave).

Per creare una KMS chiave con materiale chiave importato, inizia seguendo le istruzioni per creare una KMS chiave del tipo di chiave preferito, con la seguente eccezione.

Una volta scelto l'utilizzo della chiave, effettua le seguenti operazioni:

1. Espandere Advanced options (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 multiregionale con materiale KMS chiave importato, in Regionalità seleziona Chiave multiregionale.
5. Torna alle istruzioni basilari. I passaggi rimanenti della procedura di base sono gli stessi per tutte le KMS chiavi di quel tipo.

Quando scegliete Finish, avete creato una KMS chiave senza materiale chiave e con lo stato (stato chiave) di In attesa di importazione.

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 KMS chiave senza materiale chiave ()AWS KMS API

Per utilizzare la AWS KMS APIper creare una KMS chiave di crittografia simmetrica priva di materiale chiave, inviate una CreateKeyrichiesta 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 è `EXTERNAL` e `Origin` la sua `KeyState` è. `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 KMS chiave di crittografia simmetrica con un `KeySpec` of `SYMMETRIC_DEFAULT` e `KeyUsage` un of. `ENCRYPT_DECRYPT` È possibile utilizzare i parametri opzionali `--key-spec` e `--key-usage` creare una chiave o asimmetrica. HMAC KMS Per ulteriori informazioni, vedere 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 KMS chiave utilizzando la AWS KMS console o il. GetParametersForImportAPI 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 RSA wrapping, selezionate la lunghezza (specifiche chiave) della RSA coppia di chiavi di wrapping e l'algoritmo di wrapping che utilizzerete per crittografare il materiale chiave importato per il trasporto nella fase 3. AWS KMS supporta anche la specifica della chiave di SM2 avvolgimento (solo per le regioni cinesi).

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 RSA avvolgimento («specifica chiave») o di sostituire un set perso.

Puoi anche scaricare una chiave pubblica di avvolgimento e importare un set di token per reimportare lo stesso materiale 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 AES simmetrica, il tutto all'interno di un modulo di sicurezza hardware (). AWS KMS HSM

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 privata non li lascia mai HSM in testo semplice.

RSAspecifiche chiave del confezionamento

Le specifiche della chiave pubblica di imballaggio determinano la lunghezza delle chiavi nella coppia di chiavi che protegge il RSA materiale della chiave durante il trasporto verso. AWS KMS In generale, consigliamo di utilizzare la chiave pubblica di wrapping più lunga possibile tale che sia pratica. Offriamo diverse specifiche di chiave pubblica di imballaggio per supportare una varietà di HSMs gestori chiave.

AWS KMS supporta le seguenti specifiche chiave per le chiavi di RSA avvolgimento utilizzate per importare materiale chiave di tutti i tipi, ad eccezione di quanto indicato.

- RSA_4096 (preferito)
- RSA_3072
- RSA_2048

> **Note**
>
> È NOT supportata la seguente combinazione: materiale chiave ECC _ NIST _P521, la specifica della chiave di wrapping pubblica RSA _2048 e un algoritmo di wrapping _ _ _*. RSAES OAEP SHA
> Non è possibile avvolgere direttamente il materiale chiave ECC _ NIST _P521 con una chiave di wrapping pubblica _2048. RSA Utilizzate una chiave di avvolgimento più grande o un algoritmo di avvolgimento RSA _ _ _ _ _*AES. KEY WRAP SHA

SM2specifiche della chiave di avvolgimento (solo regioni cinesi)

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 avvolgimento standard e un algoritmo di RSA avvolgimento 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. In genere, si sceglie un algoritmo supportato dal modulo di sicurezza hardware (HSM) o dal sistema di gestione delle chiavi che protegge il materiale chiave.

La tabella seguente mostra gli algoritmi di wrapping supportati per ogni tipo di materiale e KMS chiave chiave. Gli algoritmi sono elencati in ordine di preferenza.

Materiale chiave	Specifiche e algoritmo di wrapping supportati
Chiave di crittografia simmetrica Chiave a 256 bit AES SM4Chiave a 128 bit (solo regioni cinesi)	Algoritmi di wrapping: RSAES_ OAEP _ _256 SHA RSAES_ OAEP _ _1 SHA Algoritmi di wrapping obsoleti: RSAES_ PKCS1 _V1 **Note** Al 10 ottobre 2023, non AWS KMS supporta l'algoritmo di wrapping RSAES _ PKCS1 _V1_5.

Materiale chiave	Specifiche e algoritmo di wrapping supportati
	Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096
Chiave privata asimmetrica RSA	Algoritmi di wrapping: RSA_ _ _ AES _ _256 KEY WRAP SHA RSA_ _ AES _ KEY WRAP _ _1 SHA 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 è possibile utilizzare gli algoritmi di wrapping RSAES _ OAEP _ SHA _* con la specifica della chiave di wrapping RSA _2048 per avvolgere il materiale chiave _ _P521. ECC NIST	Algoritmi di wrapping: RSASHA_ _ _ _ _256 AES KEY WRAP RSA_ _ AES _ KEY WRAP _ _1 SHA RSAES_ OAEP _ _256 SHA RSAES_ OAEP _ _1 SHA 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 cinesi)	Algoritmi di wrapping: RSAES_ _ _256 OAEP SHA RSAES_ OAEP _ _1 SHA 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
HMACchiave	Algoritmi di wrapping: RSAES_ OAEP _ SHA _256 RSAES_ OAEP _ _1 SHA 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 cinesi.

- `RSA_AES_KEY_WRAP_SHA_256`— Un algoritmo di wrapping ibrido in due fasi che combina la crittografia del materiale chiave con una chiave AES simmetrica generata dall'utente e quindi la crittografia della chiave AES simmetrica con la chiave di wrapping RSA pubblica scaricata e l'algoritmo di wrapping _ _ _256. RSAES OAEP SHA

  È necessario un algoritmo di `RSA_AES_KEY_WRAP_SHA_*` wrapping per racchiudere il materiale a chiave RSA privata, tranne che nelle regioni cinesi, dove è necessario utilizzare l'algoritmo di wrapping. `SM2PKE`
- `RSA_AES_KEY_WRAP_SHA_1`— Un algoritmo di wrapping ibrido in due fasi che combina la crittografia del materiale chiave con una chiave AES simmetrica generata dall'utente e quindi la crittografia della chiave simmetrica con la chiave pubblica di wrapping scaricata RSA e l'AESalgoritmo di wrapping _ _ _1. RSAES OAEP SHA

  È necessario un algoritmo di `RSA_AES_KEY_WRAP_SHA_*` wrapping per racchiudere il materiale a chiave RSA privata, tranne che nelle regioni cinesi, dove è necessario utilizzare l'algoritmo di wrapping. `SM2PKE`

- `RSAES_OAEP_SHA_256`— L'algoritmo di RSA crittografia con Optimal Asymmetric Encryption Padding () OAEP con la funzione hash -256. SHA
- `RSAES_OAEP_SHA_1`— L'algoritmo di RSA crittografia con Optimal Asymmetric Encryption Padding () con la funzione hash -1. OAEP SHA
- `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 con il formato di riempimento definito nella versione #1 1.5. RSA PKCS
- `SM2PKE`(Solo regioni cinesi) — Un algoritmo di crittografia basato su curve ellittiche definito in GM/T 0003.4-2012. OSCCA

Argomenti

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 i passaggi per creare una KMS chiave senza materiale chiave e ti trovi nella pagina Scarica la chiave di wrapping e importa il token, vai a. Step 9
2. Accedi a AWS Management Console 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**
>
> È possibile importare materiale chiave solo in una KMS chiave con Origine esterna (Importa materiale chiave). Ciò indica che la KMS chiave 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. Scegliete l'alias o l'ID della KMS chiave in attesa di importazione.
6. Scegli la tab Configurazione crittografica e visualizzane i valori. Le tab si trovano nella sezione Configurazione generale.

   È possibile importare solo materiale chiave nelle KMS chiavi di Origin of External (Import Key Material). Per informazioni sulla creazione di KMS chiavi con materiale chiave importato, consultate,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 chiave viene visualizzata solo per KMS le chiavi il cui valore Origin è Esterno (Importa materiale chiave).
8. Per Select wrapping key spec, scegli la configurazione per la tua KMS chiave. 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 avvolge 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 ()AWS KMS API

Per scaricare la chiave pubblica e il token di importazione, usa il GetParametersForImportAPI. Specificate la KMS chiave che verrà associata al materiale chiave importato. Questa KMS chiave deve avere un valore Origin pari a`EXTERNAL`.

Questo esempio specifica l'algoritmo di `RSA_AES_KEY_WRAP_SHA_256` wrapping, la specifica della chiave pubblica di wrapping RSA _3072 e un ID di chiave di esempio. Sostituisci questi valori di esempio con valori validi per il download. Per l'ID della chiave, è possibile utilizzare un ID o una chiave ARN, ma non è possibile utilizzare un nome alias o un alias in questa operazione. ARN

```
$ 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. Copia la chiave pubblica codificata in base64 (rappresentata da *`public key (base64 encoded)`* nell'output di esempio), incollatela in un nuovo file, quindi salvate il file. Assegna al file un nome descrittivo, ad esempio `PublicKey.b64`.
2. Usa Open SSL to base64 per decodificare 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. Copia il token di importazione codificato in base64 (rappresentato da *import token (base64 encoded)* nell'output di esempio), incollatelo in un nuovo file, quindi salvatelo. Assegna al file un nome descrittivo, ad esempio `importtoken.b64`.
4. Usa Open SSL to base64 per decodificare 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, vedere Selezione di una specifica della chiave pubblica di wrapping e. 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.
> È NOT supportata la seguente combinazione: materiale chiave ECC _ NIST _P521, la specifica della chiave di wrapping pubblica RSA _2048 e un algoritmo di wrapping _ _ _*. RSAES OAEP SHA
> Non è possibile avvolgere direttamente il materiale chiave ECC _ NIST _P521 con una chiave di wrapping pubblica _2048. RSA Utilizzate una chiave di avvolgimento più grande o un algoritmo di avvolgimento 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 cinesi.

In genere, si crittografa il materiale chiave quando lo si esporta dal modulo di sicurezza hardware (HSM) o dal sistema di gestione delle chiavi. Per informazioni su come esportare il materiale chiave in formato binario, consultate la documentazione del vostro sistema HSM di gestione delle chiavi. È inoltre possibile fare riferimento alla sezione seguente che fornisce una dimostrazione dimostrativa dell'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, vedete l'evento di CloudTrail registro per la richiesta associata GetParametersForImport.

Genera il materiale della chiave per i test

I seguenti SSL comandi Open generano materiale 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 DER -encoded, reindirizzate il comando key material generation al comando seguente. `openssl pkcs8` Il `topk8` parametro indica SSL a Open di accettare una chiave privata come input e restituire una chiave in formato #8. PKCS (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
```

- HMACchiavi

  Questo comando genera una stringa di byte casuale della dimensione specificata. Non è necessario codificare questo materiale della chiave.

  La lunghezza della HMAC chiave deve corrispondere alla lunghezza definita dalle specifiche della KMS chiave. Ad esempio, se la KMS chiave è HMAC _384, è necessario 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
```

- RSAchiavi private

```
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
```

- ECCchiavi private

```
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
```

- SM2chiavi private (solo regioni cinesi)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -
outform der -nocrypt > SM2_PrivateKey.der
```

Esempi di crittografia del materiale chiave con Open SSL

Gli esempi seguenti mostrano come utilizzare Open SSL per crittografare il materiale chiave con la chiave pubblica scaricata. Per crittografare il materiale chiave utilizzando una chiave SM2 pubblica (solo per le regioni cinesi), 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, utilizzate un metodo più sicuro (ad esempio un sistema commerciale HSM o di gestione delle chiavi) per generare e archiviare il materiale chiave.
> È NOT supportata la seguente combinazione: materiale chiave ECC _ NIST _P521, la specifica della chiave di wrapping pubblica RSA _2048 e un algoritmo di wrapping _ _ _*. RSAES OAEP SHA
> Non è possibile avvolgere direttamente il materiale chiave ECC _ NIST _P521 con una chiave di wrapping pubblica _2048. RSA Utilizzate una chiave di avvolgimento più grande o un algoritmo di avvolgimento RSA _ _ _ _ _*AES. KEY WRAP SHA

RSAES_OAEP_SHA_1

AWS KMS supporta RSAES _ OAEP _ SHA _1 per le chiavi di crittografia simmetriche (SYMMETRIC_DEFAULT), le chiavi private a curva ellittica (ECC), le chiavi private e le chiavi. SM2 HMAC

RSAES_ OAEP _ SHA _1 non è supportato per le chiavi private. RSA Inoltre, non è possibile utilizzare una chiave di wrapping pubblica RSA _2048 con alcun algoritmo di wrapping RSAES _ _ SHA _* per avvolgere una chiave privata OAEP _ _P521 (ECCsecp521r1NIST). È necessario

utilizzare una chiave di wrapping pubblica più grande o un algoritmo di wrapping _ _ _. RSA AES KEY WRAP

L'esempio seguente cripta il materiale chiave con la chiave pubblica scaricata e l'algoritmo di wrapping RSAES OAEP _ _ SHA _1 e lo salva 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 chiave da crittografare, ad esempio `PlaintextKeyMaterial.bin` o. `HMAC_384_PlaintextKey.bin` `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 le chiavi di crittografia simmetriche (SYMMETRIC_DEFAULT), le chiavi private a curva ellittica (ECC), SM2 le chiavi private e le chiavi. HMAC

RSAES_ OAEP _ SHA _256 non è supportato per le chiavi private. RSA Inoltre, non è possibile utilizzare una chiave di wrapping pubblica RSA _2048 con alcun algoritmo di wrapping RSAES _ _ SHA _* per avvolgere una chiave privata OAEP _ _P521 (ECCsecp521r1NIST). È necessario utilizzare una chiave pubblica più grande o un algoritmo di wrapping _ _ _. RSA AES KEY WRAP

L'esempio seguente cripta il materiale chiave con la chiave pubblica scaricata e l'algoritmo di wrapping RSAES OAEP _ _ SHA _256 e lo salva 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 chiave da crittografare, ad esempio `PlaintextKeyMaterial.binHMAC_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. Crittografa il materiale della chiave con una chiave AES simmetrica generata da te e un algoritmo di crittografia simmetrico. AES
2. Crittografa la chiave AES simmetrica che hai usato con la chiave pubblica che hai scaricato e l'algoritmo di wrapping _ _ _1. RSAES OAEP SHA

L'algoritmo di AES wrapping RSA _ KEY _ _ WRAP _ SHA _1 richiede la versione Open 3. SSL x o versione successiva.

1. Genera una chiave di crittografia AES simmetrica a 256 bit

   Questo comando genera una chiave di crittografia AES simmetrica 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. Crittografa il materiale chiave con la chiave di crittografia simmetrica AES

   Questo comando crittografa il materiale chiave con la chiave di crittografia AES simmetrica e salva il materiale chiave crittografato nel file. `key-material-wrapped.bin`

   In questo esempio di comando:

   - *`PlaintextKeyMaterial.bin`* è il file che contiene il materiale chiave da importare, ad esempio `PlaintextKeyMaterial.bin` `HMAC_384_PlaintextKey.binRSA_3072_PrivateKey.der`, o. `ECC_NIST_P521_PrivateKey.der`
   - *`aes-key.bin`* è il file che contiene la chiave di crittografia AES simmetrica 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. Crittografa la tua chiave di crittografia AES simmetrica con la chiave pubblica

   Questo comando cripta la chiave di crittografia AES simmetrica con la chiave pubblica scaricata e l'algoritmo di wrapping RSAES _ OAEP _ SHA _1, la DER codifica 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 AES simmetrica a 256 bit generata nel primo comando di questa sequenza di esempio.

```
# 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 chiave crittografato e il file con la chiave crittografata. AES 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 crittografata.
   - *`aes-key-wrapped.bin`* è il file che contiene la chiave di AES crittografia 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. Crittografa il materiale della chiave con una chiave AES simmetrica generata da te e un algoritmo di crittografia simmetrico. AES
2. Crittografa la chiave AES simmetrica che hai usato con la chiave pubblica che hai scaricato e l'algoritmo di wrapping _ _ _256. RSAES OAEP SHA

L'algoritmo di AES wrapping RSA _ KEY _ WRAP _ SHA _256 richiede la versione Open 3. SSL x o versione successiva.

1. Genera una chiave di crittografia AES simmetrica a 256 bit

   Questo comando genera una chiave di crittografia AES simmetrica 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. Crittografa il materiale chiave con la chiave di crittografia simmetrica AES

   Questo comando crittografa il materiale chiave con la chiave di crittografia AES simmetrica e salva il materiale chiave crittografato nel file. `key-material-wrapped.bin`

   In questo esempio di comando:

   - *`PlaintextKeyMaterial.bin`* è il file che contiene il materiale chiave da importare, ad esempio `PlaintextKeyMaterial.bin` `HMAC_384_PlaintextKey.binRSA_3072_PrivateKey.der`, o. `ECC_NIST_P521_PrivateKey.der`
   - *`aes-key.bin`* è il file che contiene la chiave di crittografia AES simmetrica 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. Crittografa la tua chiave di crittografia AES simmetrica con la chiave pubblica

   Questo comando cripta la chiave di crittografia AES simmetrica con la chiave pubblica scaricata e l'algoritmo di wrapping RSAES _ OAEP _ SHA _256, la DER codifica 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 AES simmetrica a 256 bit generata nel primo comando di questa sequenza di esempio.

```
# 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 chiave crittografato e il file con la chiave crittografata. AES 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 crittografata.
   - *aes-key-wrapped.bin* è il file che contiene la chiave di AES crittografia 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 chiave nella stessa KMS chiave specificata al momento del download della chiave pubblica e del token di importazione. Quando il materiale chiave viene importato correttamente, lo stato della KMS chiave cambia in `Enabled` ed è possibile utilizzare la KMS chiave nelle operazioni crittografiche.

Quando importi il materiale della chiave, puoi impostare un'ora di scadenza facoltativa per il materiale della chiave. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave e la KMS chiave diventa inutilizzabile. Per utilizzare la KMS chiave nelle operazioni crittografiche, è necessario reimportare lo stesso materiale chiave. Dopo aver importato il materiale della chiave, non potrai impostare, modificare o annullare la data di scadenza dell'importazione corrente. Per modificare questi valori, dovrai eliminare e reimportare lo stesso materiale della chiave.

Per importare materiale chiave, è possibile utilizzare la AWS KMS console o il. ImportKeyMaterialAPI È possibile API utilizzarlo direttamente effettuando HTTP richieste, oppure utilizzando un AWS SDKs, AWS Command Line Interfaceo AWS Tools for PowerShell.

Quando si importa il materiale chiave, viene aggiunta una ImportKeyMaterialvoce al AWS CloudTrail registro per registrare l'`ImportKeyMaterial`operazione. La CloudTrail voce è la stessa sia che si utilizzi la AWS KMS console che il AWS KMS API.

Impostazione di una data di scadenza (facoltativo)

Quando importi il materiale chiave per la tua KMS chiave, puoi impostare una data e un'ora di scadenza facoltative per il materiale chiave fino a 365 giorni dalla data di importazione. Quando il materiale chiave importato scade, lo AWS KMS elimina. Questa azione modifica lo stato della KMS chiave in`PendingImport`, impedendone l'utilizzo in qualsiasi operazione crittografica. Per utilizzare la KMS chiave, è necessario reimportare una copia del materiale della chiave originale.

Garantire che il materiale chiave importato scada frequentemente può aiutare a soddisfare i requisiti normativi, ma comporta un ulteriore rischio per i dati crittografati con la chiave. KMS Fino a quando non si reimporta una copia del materiale chiave originale, una chiave con materiale KMS chiave scaduto è inutilizzabile e tutti i dati crittografati sotto la chiave sono inaccessibili. KMS Se non

si riesce a reimportare il materiale chiave per qualsiasi motivo, inclusa la perdita della copia del materiale chiave originale, la chiave è definitivamente inutilizzabile e i dati crittografati sotto la KMS chiave non sono recuperabili. KMS

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 chiaveimportato, nonché l'operazione per eliminare il AWS KMS materiale chiave scaduto.

Non è possibile importare materiale chiave diverso nella KMS chiave e 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, puoi attivare e disattivare la scadenza o impostare una nuova data di scadenza eliminando e reimportando il materiale della chiave. Utilizzate il `ExpirationModel` parametro di ImportKeyMaterialper 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.

Reimporta il materiale chiave

Se gestite una KMS chiave con materiale chiave importato, potrebbe essere necessario reimportare il materiale 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.

Quando importate del materiale chiave in una KMS chiave, la KMS chiave viene associata in modo permanente a quel materiale chiave. È possibile reimportare lo stesso materiale chiave, ma non è possibile importare materiale chiave diverso in quella KMS chiave. Non è possibile ruotare il materiale chiave e AWS KMS non è possibile creare materiale chiave per una KMS chiave con materiale chiave importato.

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.

Le procedure per reimportare il materiale chiave sono le stesse utilizzate per importare il materiale chiave per la prima volta, con le seguenti eccezioni.

- Utilizzate una KMS chiave esistente, invece di crearne una nuovaKMS. Puoi saltare la fase 1 della procedura di importazione.
- Quando si reimporta il materiale della chiave, è possibile modificare il modello di scadenza e la data di scadenza.

Ogni volta che importi materiale chiave in una KMS chiave, devi scaricare e utilizzare una nuova chiave di wrapping e importare il token per la KMS chiave. 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.

Importazione del materiale della chiave (console)

È possibile utilizzare il AWS Management Console per importare materiale chiave.

1. Se ti trovi nella pagina Carica il materiale della chiave di wrapping, passa a Step 8.
2. Accedi a AWS Management Console 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'ID o l'alias della chiave per la quale hai scaricato la KMS chiave pubblica e il token di importazione.
6. Scegli la tab Configurazione crittografica e visualizzane i valori. Le schede si trovano nella pagina dei dettagli di una KMS chiave sotto la sezione Configurazione generale.

   È possibile importare solo materiale chiave in KMS chiavi con Origine esterna (Importa materiale chiave). Per informazioni sulla creazione di KMS chiavi con materiale chiave importato, vedereImportazione 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 chiave viene visualizzata solo per KMS le chiavi con un valore Origin uguale a Esterno (Importa materiale chiave).

   Se hai scaricato il materiale della chiave, il token di importazione e hai crittografato il materiale della chiave, scegli Avanti.

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 chiave (crittografato) sottoposto a wrapping.

   b. In Token di importazione, sceglie 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. Scegliere Upload key material (Carica materiale chiave).

Importazione di materiale della chiave (AWS KMS API)

Per importare materiale chiave, utilizzate l'ImportKeyMaterialoperazione. Gli esempi seguenti utilizzano la AWS CLI, ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Per utilizzare questo esempio:

1. Sostituisci *1234abcd-12ab-34cd-56ef-1234567890ab* con un ID chiave della KMS chiave che hai specificato quando hai scaricato la chiave pubblica e il token di importazione. Per identificare la KMS chiave, usa il relativo ID o chiave ARN. Non è possibile utilizzare un nome alias o un alias ARN per questa operazione.

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
```

> **Tip**
>
> Se l'esito del comando non è positivo, potresti visualizzare un'`KMSInvalidStateException` o un'`NotFoundException`. Puoi ritentare la richiesta.

Creare una KMS chiave in un archivio di AWS CloudHSM chiavi

Dopo aver creato un archivio AWS CloudHSM chiavi, puoi crearlo AWS KMS keys nel tuo archivio chiavi. Devono essere KMSchiavi di crittografia simmetriche con materiale chiave generato AWS KMS . Non è possibile creare chiavi asimmetriche, KMS chiavi o HMACKMSKMSchiavi con materiale chiave importato in un archivio di chiavi personalizzato. Inoltre, non è possibile utilizzare chiavi di crittografia simmetriche in un archivio di KMS chiavi personalizzato per generare coppie di chiavi di dati asimmetriche.

Per creare una KMS chiave in un archivio di AWS CloudHSM chiavi, 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 connessioniHSMs, visualizza la pagina degli archivi di AWS CloudHSM chiavi nel AWS Management Console. Quando si utilizzano le API operazioni, utilizzare l'DescribeCustomKeyStoresoperazione

per verificare che l'archivio 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 DescribeClustersoperazione.

Quando crei una KMS chiave nel tuo archivio AWS CloudHSM chiavi, AWS KMS crea la KMS chiave in AWS KMS. Tuttavia, crea il materiale chiave per la KMS chiave nel AWS CloudHSM cluster associato. In particolare AWS KMS , accede al cluster come `kmsuser`CU che hai creato. Quindi crea una chiave simmetrica Advanced Encryption Standard (AES) persistente, non estraibile, a 256 bit nel cluster. AWS KMS imposta il valore dell'attributo key label, visibile solo nel cluster, su Amazon Resource Name (ARN) della KMS chiave.

Quando il comando ha esito positivo, lo stato chiave della nuova KMS chiave è `Enabled` e la sua origine è`AWS_CLOUDHSM`. Non è possibile modificare l'origine di alcuna KMS chiave dopo averla creata. Quando si visualizza una KMS chiave in un archivio AWS CloudHSM chiavi della AWS KMS console o utilizzando l'DescribeKeyoperazione, 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 KMS chiave 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 KMS chiave in un archivio di AWS CloudHSM chiavi, vedereCreateKey.

Crea una nuova KMS chiave nel tuo archivio di HSM chiavi Cloud

Puoi creare una KMS chiave di crittografia simmetrica nel tuo archivio di AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'CreateKeyoperazione.

Utilizzo della console AWS KMS

Utilizzare la procedura seguente per creare una KMS chiave di crittografia simmetrica in un archivio di AWS CloudHSM 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 AWS Management Console 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. 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 KMS chiave. 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, consultaDisconnetti un archivio di AWS CloudHSM chiavi. Per informazioni sull'aggiuntaHSMs, consulta Aggiungere un file HSM nella Guida AWS CloudHSM per l'utente.
11. Scegli Next (Successivo).
12. Digita un alias e una descrizione opzionale per la KMS chiave.
13. (Facoltativo). Nella pagina Aggiungi tag, aggiungi i tag che identificano o classificano la tua KMS chiave.

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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

14. Scegli Next (Successivo).
15. Nella sezione Amministratori chiave, seleziona IAM gli utenti e i ruoli che possono gestire la KMS chiave. Per ulteriori informazioni, consulta Consente agli amministratori chiave di amministrare la chiave. KMS

> **Note**
>
> IAMle politiche possono concedere ad altri IAM utenti e ruoli il permesso di utilizzare la KMS chiave.
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> 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.

16. (Facoltativo) Per impedire a questi amministratori chiave di eliminare questa KMS chiave, deseleziona la casella in fondo alla pagina relativa a Consenti agli amministratori chiave di eliminare questa chiave.
17. Scegli Next (Successivo).
18. Nella sezione Questo account, seleziona IAM gli utenti e i ruoli Account AWS che possono utilizzare la KMS chiave nelle operazioni crittografiche. Per ulteriori informazioni, consulta Consente agli utenti chiave di utilizzare la KMS chiave.

> **Note**
>
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.
> La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori delle istruzioni `"Allow use of the key"` 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 KMS chiave 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 KMS chiave creando IAM politiche per i propri utenti. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

20. Scegli Next (Successivo).
21. Esamina 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 KMS chiave nell'archivio AWS CloudHSM chiavi che hai scelto. Quando si sceglie il nome o l'alias della nuova KMS chiave, la scheda Configurazione crittografica nella relativa pagina di dettaglio mostra l'origine della KMS chiave (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 semplificare l'identificazione KMS delle chiavi in un archivio di chiavi personalizzato, nella pagina Customer managed keys, aggiungi la colonna Customer key store ID al display. 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, consultare Personalizza la visualizzazione della console.

Utilizzando il AWS KMS API

Per creare una nuova AWS KMS key (KMSchiave) nel tuo archivio AWS CloudHSM chiavi, usa l'CreateKeyoperazione. Utilizza il parametro `CustomKeyStoreId` 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 policy 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'DescribeCustomKeyStoresoperazione 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 `CustomKeyStoreName` parametro `CustomKeyStoreId` 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": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
```

```
}
```

Il comando di esempio successivo utilizza l'DescribeClustersoperazione 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-abcdefghijk",
                    "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-zyxwvutsrqp",
                    "State": "ACTIVE"
```

```
                },
            ],
            "State": "ACTIVE"
        }
    ]
}
```

Questo comando di esempio utilizza l'CreateKeyoperazione per creare una KMS chiave in un archivio di AWS CloudHSM chiavi. Per creare una KMS chiave 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 IDs l'archivio 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",
    "CustomKeyStoreId": "cks-1234567890abcdef0"
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Creare una KMS chiave in archivi di chiavi esterni

Dopo aver creato e collegato l'archivio chiavi esterno, puoi crearlo AWS KMS keys nel tuo archivio chiavi. Devono essere KMSchiavi di crittografia simmetriche con un valore di origine di External key store ()EXTERNAL_KEY_STORE. Non è possibile creare chiavi asimmetriche, KMS chiavi o HMACKMSKMSchiavi con materiale chiave importato in un archivio di chiavi personalizzato. Inoltre, non è possibile utilizzare chiavi di crittografia simmetriche in un archivio di KMS chiavi personalizzato per generare coppie di chiavi di dati asimmetriche.

Una KMS chiave in un archivio di chiavi esterno potrebbe avere una latenza, una durata e una disponibilità inferiori rispetto a una KMS chiave standard perché dipende da componenti situati all'esterno di. AWS Prima di creare o utilizzare una KMS chiave in un archivio di chiavi esterno, verificate di aver bisogno di una chiave con proprietà di archivio chiavi esterne.

> **Note**
>
> Alcuni gestori di chiavi esterni forniscono un metodo più semplice per creare KMS chiavi in un archivio di chiavi esterno. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

Per creare una KMS chiave nell'archivio di chiavi esterno, specificate 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 chiave per la KMS chiave. Non è possibile modificare l'ID della chiave esterna dopo aver creato la KMS chiave.

  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 KMS chiave in un archivio di chiavi esterno contiene anche materiale AWS KMS chiave. Tutti i dati crittografati con la KMS chiave vengono prima crittografati AWS KMS utilizzando il materiale AWS KMS chiave della chiave e poi dal gestore delle chiavi esterno

utilizzando la chiave esterna. Questo processo di doppia crittografia garantisce che il testo cifrato protetto da una KMS chiave in un archivio di chiavi esterno sia almeno altrettanto potente del testo cifrato protetto solo da. AWS KMS Per informazioni dettagliate, consultare Funzionamento degli archivi delle chiavi esterne.

Quando l'`CreateKey`operazione ha esito positivo, lo stato della nuova chiave è. KMS `Enabled` Quando si visualizza una KMS chiave in un archivio di chiavi esterno, è possibile visualizzare le proprietà tipiche, come l'ID della chiave, le specifiche della chiave, 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 KMS chiave nell'archivio di chiavi esterno fallisce, utilizza il messaggio di errore per identificare la causa. Potrebbe indicare che l'archivio chiavi esterno non è connesso (`CustomKeyStoreInvalidStateException`), che il proxy dell'archivio chiavi esterno non riesce a trovare una chiave esterna con l'ID chiave esterna specificato (`XksKeyNotFoundException`) o che la chiave esterna è già associata a una KMS chiave nello stesso archivio di chiavi esterno`XksKeyAlreadyInUseException`.

Per un esempio del AWS CloudTrail registro dell'operazione che crea una KMS chiave in un archivio di chiavi esterno, vedereCreateKey.

Argomenti

- Requisiti per una KMS chiave in un archivio di chiavi esterno
- Crea una nuova KMS chiave nel tuo archivio di chiavi esterno

Requisiti per una KMS chiave in un archivio di chiavi esterno

Per creare una KMS chiave in un archivio di chiavi esterno, sono necessarie le seguenti proprietà dell'archivio chiavi esterno, della KMS chiave e della chiave esterna che funge da materiale crittografico esterno per la KMS chiave.

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.

KMSrequisiti chiave

Non è possibile modificare queste proprietà dopo aver creato la KMS chiave.

- Specifiche chiave: _ SYMMETRIC DEFAULT
- Utilizzo della chiave: _ ENCRYPT DECRYPT
- Origine del materiale chiave: EXTERNAL _ KEY _ STORE
- Multiregione: FALSE

Requisiti della chiave esterna

- Chiave AES crittografica 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 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 preparare chiavi esterne per gestire fino a 1800 richieste al secondo.

Crea una nuova KMS chiave nel tuo archivio di chiavi esterno

È possibile creare una nuova KMS chiave nell'archivio delle chiavi esterno nella AWS KMS console o utilizzando l'CreateKeyoperazione.

Utilizzo della AWS KMS console

Esistono due modi per creare una KMS chiave in un archivio chiavi esterno.

- Metodo 1 (consigliato): scegli un archivio chiavi esterno, quindi crea una KMS chiave in quell'archivio di chiavi esterno.
- Metodo 2: crea una KMS chiave, quindi indica che si trova in un archivio di chiavi esterno.

Se utilizzate il Metodo 1, in cui scegliete l'archivio di chiavi esterno prima di creare la chiave, AWS KMS scegliete automaticamente tutte le proprietà della KMS chiave richieste e inserite l'ID del vostro archivio di chiavi esterno. Questo metodo evita errori che potreste commettere 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 il tuo archivio di chiavi esterno, quindi crea una KMS chiave. 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 potreste commettere durante la creazione della chiave. KMS

1. Accedi AWS Management Console 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. Scegli il nome dell'archivio delle chiavi esterne.
5. Nell'angolo in alto a destra, scegli Crea una KMS chiave in questo archivio di chiavi.

   Se l'archivio delle chiavi esterne non è connesso, ti verrà richiesto di collegarlo. Se il tentativo di connessione fallisce, devi risolvere il problema e connettere l'archivio di chiavi esterno prima di potervi creare una nuova KMS chiave.

   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'utilizzo con una KMS chiave. Non puoi modificare questo valore dopo la creazione della chiave.

   Se la chiave esterna è multiplaIDs, inserisci l'ID della chiave utilizzato dal proxy dell'archivio chiavi esterno per identificare la chiave esterna.

7. Conferma che intendi creare una KMS chiave nell'archivio di chiavi esterno specificato.
8. Scegli Next (Successivo).

   Il resto di questa procedura equivale alla creazione di una KMS chiave standard.

9. Digitate un alias (obbligatorio) e una descrizione (opzionale) per la KMS chiave.
10. (Facoltativo). Nella pagina Aggiungi tag, aggiungi i tag che identificano o classificano la tua KMS chiave.

    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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

11. Scegli Next (Successivo).
12. Nella sezione Amministratori chiave, seleziona IAM gli utenti e i ruoli che possono gestire la KMS chiave. Per ulteriori informazioni, consulta Consente agli amministratori chiave di amministrare la chiave. KMS

    > **Note**
    >
    > IAMle politiche possono concedere ad altri IAM utenti e ruoli il permesso di utilizzare la KMS chiave.
    > IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

13. (Facoltativo) Per impedire a questi amministratori chiave di eliminare questa KMS chiave, deseleziona la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.

L'eliminazione di una KMS chiave è un'operazione distruttiva e irreversibile che può rendere irrecuperabile il testo cifrato. Non è possibile ricreare una chiave simmetrica in un archivio di chiavi esterno, anche se si dispone del materiale KMS chiave esterno. Tuttavia, l'eliminazione di una KMS chiave non ha alcun effetto sulla chiave esterna associata. Per informazioni sull'eliminazione di una KMS chiave da un archivio chiavi esterno, vedere Considerazioni speciali per l'eliminazione delle chiavi.

14. Scegli Next (Successivo).
15. Nella sezione Questo account, seleziona IAM gli utenti e i ruoli Account AWS che possono utilizzare la KMS chiave nelle operazioni crittografiche. Per ulteriori informazioni, consulta Consente agli utenti chiave di utilizzare la KMS chiave.

> **Note**
>
> IAMle politiche possono concedere ad altri IAM utenti e ruoli il permesso di utilizzare la KMS chiave.
> IAMle migliori pratiche scoraggiano l'uso di IAM utenti con credenziali a lungo termine. Quando possibile, utilizzate IAM ruoli che forniscono credenziali temporanee. Per i dettagli, consulta le migliori pratiche di sicurezza IAM nella Guida per l'IAMutente.

16. (Facoltativo) È possibile consentire Account AWS ad altri di utilizzare questa KMS chiave 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 KMS chiave creando IAM politiche per i propri utenti. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

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 AWS KMS materiale chiave. 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. È inoltre necessario specificare i valori delle proprietà richiesti per una KMS chiave in un archivio di chiavi esterno, ad esempio le specifiche della chiave e l'utilizzo della chiave.

1. Accedi a AWS Management Console 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. 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 KMS chiave nell'archivio chiavi esterno specificato.
10. Scegli Next (Successivo).
11. Scegliete la riga che rappresenta l'archivio di chiavi esterno per la nuova KMS chiave.

    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, consultare 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'utilizzo con una KMS chiave. Non puoi modificare questo valore dopo la creazione della chiave.

    Se la chiave esterna è multiplaIDs, 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 equivale alla creazione di una KMS chiave standard.

14. Digitate un alias e una descrizione opzionale per la KMS chiave.
15. (Facoltativo). Nella pagina Aggiungi tag, aggiungi i tag che identificano o classificano la tua KMS chiave.

    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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.
16. Scegli Next (Successivo).
17. Nella sezione Amministratori chiave, seleziona IAM gli utenti e i ruoli che possono gestire la KMS chiave. Per ulteriori informazioni, consulta Consente agli amministratori chiave di amministrare la chiave. KMS

    > **Note**
    >
    > IAMle politiche possono concedere ad altri IAM utenti e ruoli il permesso di utilizzare la KMS chiave.
18. (Facoltativo) Per impedire a questi amministratori chiave di eliminare questa KMS chiave, deselezionate la casella di controllo Consenti agli amministratori chiave di eliminare questa chiave.

    L'eliminazione di una KMS chiave è un'operazione distruttiva e irreversibile che può rendere irrecuperabile il testo cifrato. Non è possibile ricreare una chiave simmetrica in un archivio di chiavi esterno, anche se si dispone del materiale KMS chiave esterno. Tuttavia, l'eliminazione di una KMS chiave non ha alcun effetto sulla chiave esterna associata. Per informazioni sull'eliminazione di una KMS chiave da un archivio chiavi esterno, vedere. Eliminare un AWS KMS keys
19. Scegli Next (Successivo).
20. Nella sezione Questo account, seleziona gli IAM utenti e i ruoli Account AWS che possono utilizzare la KMS chiave nelle operazioni crittografiche. Per ulteriori informazioni, consulta Consente agli utenti chiave di utilizzare la KMS chiave.

> **Note**
>
> IAMle politiche possono concedere ad altri IAM utenti e ruoli il permesso di utilizzare la KMS chiave.

21. (Facoltativo) È possibile consentire Account AWS ad altri di utilizzare questa KMS chiave 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 KMS chiave creando IAM politiche per i propri utenti. Per ulteriori informazioni, consulta Consentire agli utenti di altri account di utilizzare una KMS chiave.

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 ha esito positivo, il display mostra la nuova KMS chiave nell'archivio di chiavi esterno scelto. Quando si sceglie il nome o l'alias della nuova KMS chiave, la scheda Configurazione crittografica nella relativa pagina dei dettagli mostra l'origine della KMS chiave (archivio chiavi esterno), il nome, l'ID e il tipo dell'archivio chiavi personalizzato, 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 semplificare l'identificazione KMS delle chiavi in un archivio di chiavi personalizzato, nella pagina Customer managed keys, aggiungi la colonna Origin and Custom key store ID al display. Per modificare i campi della tabella, scegli l'icona a forma di ingranaggio

nell'angolo in alto a destra della pagina. Per informazioni dettagliate, consultare Personalizza la visualizzazione della console.

Usando il AWS KMS API

Per creare una nuova KMS chiave in un archivio di chiavi esterno, utilizzare l'CreateKeyoperazione. 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. Questa chiave esterna deve soddisfare i requisiti per l'associazione con una KMS chiave.

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'CreateKeyoperazione per creare una KMS chiave in un archivio di chiavi esterno. La risposta include le proprietà delle KMS chiavi, l'ID dell'archivio chiavi esterno 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 AWS Management Consoleo il AWS Key Management Service (AWS KMS) API per visualizzare AWS KMS keys in ogni account e regione, comprese KMS le chiavi che gestisci e KMS le chiavi gestite da AWS.

Argomenti

Trova l'ID e la chiave della chiave ARN

Per identificarne uno AWS KMS key, puoi utilizzare l'ID della chiave o l'Amazon Resource Name (chiave ARN). Nelle operazioni crittografiche, puoi anche utilizzare il nome o l'alias dell'alias. ARN

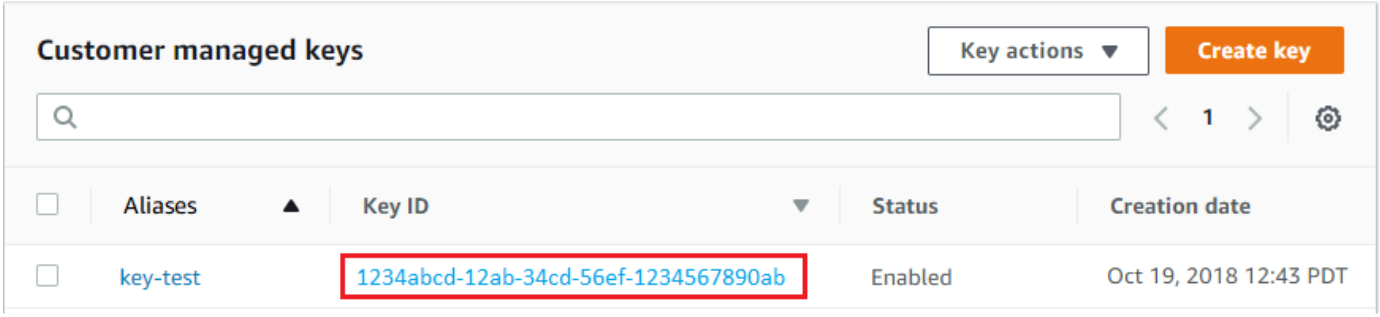
Puoi utilizzare la AWS KMS console o l'ListKeysoperazione per identificare l'ID della chiave e la chiave ARN di ogni KMS chiave nel tuo account e nella tua regione.

Per informazioni dettagliate sugli identificatori di KMS chiave supportati da AWS KMS, vedereIdentificatori chiave () KeyId. Per informazioni su come trovare il nome e l'alias di un aliasARN, consulta. Trova il nome dell'alias e l'alias ARN per una chiave KMS

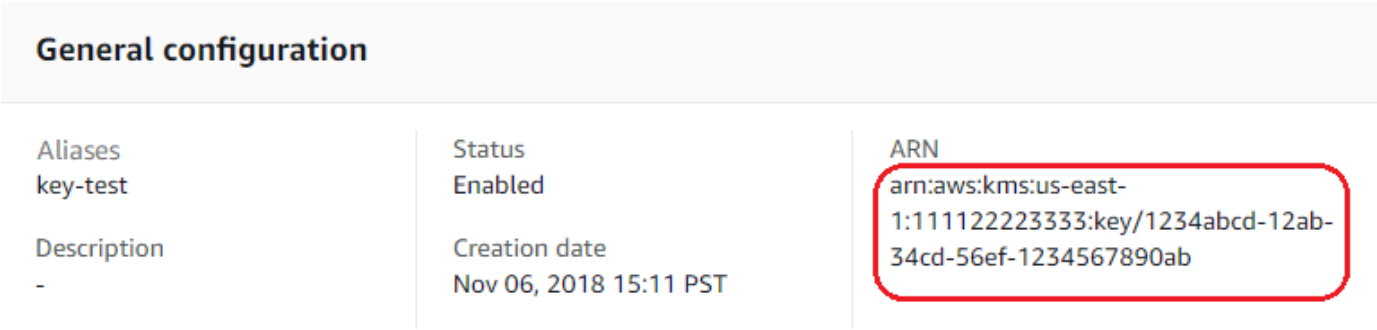
Utilizzo della console AWS KMS

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 della KMS chiave, consulta la riga che inizia con l'alias della KMS chiave.

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. Puoi anche visualizzare l'ID di una KMS chiave nella relativa pagina dei dettagli.



5. Per trovare l'Amazon Resource Name (ARN) della KMS chiave, scegli l'ID o l'alias della chiave. La chiave ARN viene visualizzata nella sezione Configurazione generale.



Utilizzando il AWS KMS API

Per trovare l'ID della chiave e la chiave ARN di an AWS KMS key, usa l'ListKeysoperazione.

L'`ListKeys`operazione restituisce l'ID della chiave e l'Amazon Resource Name (ARN) di tutte le KMS chiavi nell'account e nella regione del chiamante.

Ad esempio, questa chiamata all'`ListKeys`operazione restituisce l'ID e ARN di ogni KMS chiave in questo account fittizio. Per esempi in più linguaggi di programmazione, consulta Utilizzare `ListKeys` con un AWS SDK o 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 KMS chiave

È possibile utilizzare la AWS KMS console o l'DescribeKeyoperazione per accedere ed elencare informazioni dettagliate sulle KMS chiavi nell'account e nella regione.

Le seguenti procedure mostrano come accedere ai dettagli KMS chiave, come l'ID della chiave, le specifiche della chiave, l'utilizzo della chiave e altro ancora.

Utilizzo della console AWS KMS

La pagina dei dettagli di ogni KMS chiave mostra le proprietà della KMS chiave. Si differenzia leggermente per i diversi tipi di KMS tasti.

Per visualizzare informazioni dettagliate su una KMS chiave, nella pagina Chiavi gestite da AWS o Chiavi gestite dal cliente, scegli l'alias o l'ID della KMS chiave.

La pagina dei dettagli di una KMS chiave include una sezione di configurazione generale che mostra le proprietà di base della KMS chiave. Include anche schede in cui è possibile visualizzare e modificare le proprietà della KMS chiave, ad esempio Politica chiave, Configurazione crittografica, Tag, Materiale chiave (per KMS chiavi con materiale chiave importato), Rotazione chiave (per chiavi di crittografia simmetriche), Regionalità (per KMS chiavi multiregionali) e Chiave pubblica (per chiavi asimmetriche). KMS

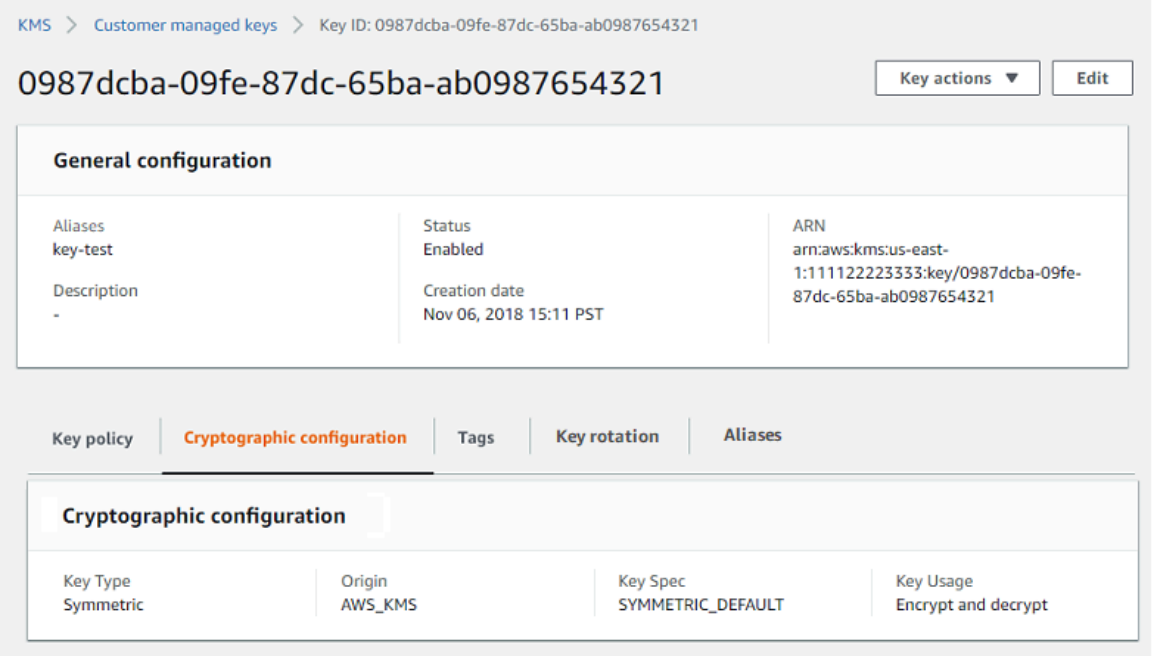
> **Note**
>
> La AWS KMS console mostra le KMS chiavi che sei autorizzato a visualizzare nel tuo account e nella tua regione. KMSle chiavi in altro Account AWS non vengono visualizzate nella

console, anche se sei autorizzato a visualizzarle, gestirle e utilizzarle. Per visualizzare KMS le chiavi in altri account, usa l'DescribeKeyoperazione.

Per accedere alla pagina dei dettagli chiave di una KMS chiave.

1. Accedi AWS Management Console 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. 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 della chiave, nella tabella delle chiavi, scegli l'ID o l'alias della KMS chiave.

   Se la KMS chiave ha più alias, accanto al nome di uno degli alias viene visualizzato un riepilogo degli alias (+ n more). La scelta del riepilogo alias consente di accedere direttamente alla sezione Alias nella pagina dei dettagli delle chiavi.



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

Un nome descrittivo per la chiave. KMS È possibile utilizzare un alias per identificare la KMS chiave nella console e in alcune altre AWS KMS APIs. Per informazioni dettagliate, consultare Alias in AWS KMS.

La scheda Alias mostra tutti gli alias associati alla KMS chiave nella Account AWS regione and.

ARN

Dove: sezione Configurazione generale

L'Amazon Resource Name (ARN) della KMS chiave. Questo valore identifica in modo univoco la KMS chiave. È possibile utilizzarlo per identificare la KMS chiave nelle AWS KMS API operazioni.

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 KMS chiave viene creata in un archivio chiavi personalizzato.

Per informazioni sui valori in questo campo, vedere ConnectionStatenel AWS KMS APIRiferimento.

Data di creazione

Dove: sezione Configurazione generale

La data e l'ora di creazione della KMS chiave. Questo valore viene visualizzato nell'ora locale per il dispositivo. Il fuso orario non dipende dalla regione.

A differenza della scadenza, la creazione si riferisce solo alla KMS chiave, non al suo materiale chiave.

ID HSM del cluster cloud

Dove: scheda Configurazione crittografica

L'ID del AWS CloudHSM cluster che contiene il materiale chiave per la KMS chiave. Questo campo viene visualizzato solo quando la KMS chiave viene creata in un archivio di chiavi personalizzato.

Se scegli l'ID del HSM cluster Cloud, viene aperta la pagina Cluster nella AWS CloudHSM console.

ID dello store delle chiavi personalizzate

Dove: scheda Configurazione crittografica

L'ID dell'archivio di chiavi personalizzato che contiene la KMS chiave. Questo campo viene visualizzato solo quando la KMS chiave viene creata in un archivio chiavi personalizzato.

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 chiavi personalizzato che contiene la KMS chiave. Questo campo viene visualizzato solo quando la KMS chiave viene creata in un archivio chiavi personalizzato.

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 KMS chiave viene creata in un archivio chiavi personalizzato.

Descrizione

Dove: sezione Configurazione generale

Una breve descrizione facoltativa della KMS chiave che è possibile 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 KMS chiave 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, vedere SYMMETRIC_ specifiche DEFAULT chiave e. RSAspecifiche chiave per la crittografia e la decrittografia

Data di scadenza

Dove: scheda Materiale della chiave

La data e l'ora di scadenza del materiale chiave per la KMS chiave. Questo campo viene visualizzato solo per KMS le chiavi con materiale chiave importato, ovvero quando l'origine è esterna e la KMS chiave ha un materiale chiave scaduto.

ID chiave esterna

Dove: scheda Configurazione crittografica

L'ID della chiave esterna associata a una KMS chiave in un archivio di chiavi esterno. Questo campo viene visualizzato solo per KMS le chiavi in un archivio di chiavi esterno.

Stato della chiave esterna

Dove: scheda Configurazione crittografica

Lo stato più recente riportato dal proxy dell'archivio chiavi esterno per la chiave esterna associata alla KMS chiave. Questo campo viene visualizzato solo per KMS le chiavi in un archivio di chiavi esterno.

Utilizzo della chiave esterna

Dove: scheda Configurazione crittografica

Le operazioni crittografiche abilitate sulla chiave esterna associata alla KMS chiave. Questo campo viene visualizzato solo per KMS le chiavi in un archivio di chiavi esterno.

Policy della chiave

Dove: scheda Policy delle chiavi

Controlla l'accesso alla KMS chiave insieme alle IAMpolitiche e alle concessioni. Ogni KMS chiave ha una politica chiave. 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, consultare the section called “Policy delle chiavi”.

Rotazione delle chiavi

Dove: Scheda Rotazione della chiave

Abilita e disabilita la rotazione automatica del materiale chiave in una KMSchiave gestita dal cliente. Per modificare lo stato di rotazione della chiave di una chiave gestita dal cliente, utilizzare la casella di controllo nella scheda Key rotation (Rotazione della chiave).

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.

Specifica della chiave

Dove: scheda Configurazione crittografica

Il tipo di materiale chiave contenuto nella KMS chiave. AWS KMS supporta KMS chiavi di crittografia simmetriche (SYMMETRIC_DEFAULT), HMAC KMS chiavi di diversa lunghezza, KMS chiavi per chiavi di diversa lunghezza e RSA chiavi a curva ellittica con curve diverse. Per informazioni dettagliate, consultare Key spec.

Tipo di chiavi

Dove: scheda Configurazione crittografica

Indica se la chiave è simmetrica o asimmetrica. KMS

Utilizzo delle chiavi

Dove: scheda Configurazione crittografica

Indica se una KMS chiave può essere utilizzata per crittografare e decrittografare, firmare e verificare o generare e verificare. MAC Per informazioni dettagliate, consultare Key usage.

Origin

Dove: scheda Configurazione crittografica

La fonte del materiale chiave per la chiave. KMS I valori validi sono:

- AWS KMS per il materiale della chiave generato da AWS KMS
- AWS CloudHSMper KMS le chiavi nel AWS CloudHSM Key Store
- Esterno per materiale chiave importato (BYOK)
- Archivio chiavi esterno per KMS chiavi in un archivio chiavi esterno

MACalgoritmi

Dove: scheda Configurazione crittografica

Elenca gli MAC algoritmi che possono essere utilizzati con una HMAC KMS chiave in. AWS KMS Questo campo viene visualizzato solo quando la specifica chiave è una specifica HMAC chiave (HMAC_*). Per informazioni sugli MAC algoritmi supportati AWS KMS , vedere. Specifiche chiave delle chiavi HMAC KMS

Chiave primaria

Dove: scheda Regionalità

Indica che questa KMS chiave è una chiave primaria multiregionale. 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 KMS chiave è una chiave primaria multiregionale.

Chiavi pubbliche

Dove: scheda Chiave pubblica

Visualizza la chiave pubblica di una chiave asimmetricaKMS. 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 KMS chiave è una chiave a regione singola, una chiave primaria multiarea o una chiave di replica multiarea. Questo campo viene visualizzato solo quando la KMS chiave è una chiave multiregionale.

Chiavi multi-regione correlate

Dove: scheda Regionalità

Visualizza tutte le chiavi primarie e di replica multiregione correlate, ad eccezione della chiave corrente. KMS Questo campo viene visualizzato solo quando la KMS chiave è una chiave multiregionale.

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 KMS chiave è una chiave di replica multiregionale. Questo campo viene visualizzato solo quando la KMS chiave è una chiave di replica multiregionale.

Algoritmi di firma

Dove: scheda Configurazione crittografica

Elenca gli algoritmi di firma che possono essere utilizzati con la KMS chiave 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 RSAspecifiche principali per la firma e la verifica e. Specifiche della chiave basata su curva ellittica

Stato

Dove: sezione Configurazione generale

Lo stato chiave della KMS chiave. È possibile utilizzare la KMS chiave nelle operazioni crittografiche solo quando lo stato è Abilitato. Per una descrizione dettagliata dello stato di ogni KMS chiave e del suo effetto sulle operazioni che è possibile eseguire sulla KMS chiave, vedereStati chiave delle AWS KMS chiavi.

Tag

Dove: scheda Tag

Coppie chiave-valore opzionali che descrivono la KMS chiave. Per aggiungere o modificare i tag per una KMS chiave, 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 KMS chiave. Per informazioni sull'etichettatura delle KMS chiavi, consulta Tag in AWS KMS eABACper AWS KMS.

Usando il AWS KMS API

L'DescribeKeyoperazione restituisce dettagli sulla KMS chiave specificata. Per identificare la KMS chiave, utilizzare l'ID della chiave ARN, il nome dell'alias o l'alias ARN.

A differenza dell'ListKeysoperazione, che visualizza solo KMS le chiavi dell'account e della regione del chiamante, gli utenti autorizzati possono utilizzare l'`DescribeKey`operazione per ottenere dettagli sulle KMS chiavi di 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 `DescribeKey` restituisce informazioni su una chiave di crittografia simmetrica. KMS 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 Utilizzare `DescribeKey` con un AWS SDK o 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"
        ]
    }
}
```

Questo esempio richiama `DescribeKey` l'operazione su una KMS chiave asimmetrica utilizzata per la firma e la verifica. La risposta include gli algoritmi di firma che AWS KMS supportano 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 DescribeKeyrisposte.

Per informazioni sulla navigazione nella scheda Configurazione crittografica nella pagina dei dettagli di una KMS chiave, consulta. the section called “Accedi ed elenca i dettagli KMS chiave”

Argomenti

- Identifica le chiavi asimmetriche KMS
- Identifica le chiavi HMAC KMS
- Identifica le chiavi multiregionali KMS
- Identifica KMS le chiavi con materiale chiave importato
- Identifica KMS le chiavi negli archivi AWS CloudHSM delle chiavi
- Identifica KMS le chiavi negli archivi di chiavi esterni

Identifica le chiavi asimmetriche KMS

Nella console AWS KMS

La colonna Tipo di chiave della tabella Customer managed keys mostra se ogni KMS chiave è simmetrica o asimmetrica. È possibile filtrare la tabella in base al valore del tipo di chiave per visualizzare solo le chiavi asimmetriche. KMS Per ulteriori informazioni, consulta the section called “Ordina e filtra le tue KMS chiavi”.

La scheda Configurazione crittografica nella pagina dei dettagli di una KMS chiave mostra il Tipo di chiave, che indica se la chiave è simmetrica o asimmetrica. Visualizza anche l'utilizzo della chiave, che indica se la chiave asimmetrica viene utilizzata per la crittografia e la decrittografia, KMS la firma e la verifica o la derivazione di segreti condivisi.

Nelle risposte DescribeKey

Quando si richiama l'`DescribeKey`operazione su una KMS chiave asimmetrica, la risposta include i `KeyUsage` valori `KeySpec` and, che possono essere utilizzati per determinare se una KMS chiave è simmetrica o asimmetrica.

Se il `KeySpec` valore è, la chiave è una chiave di crittografia `SYMMETRIC_DEFAULT` simmetrica. KMS Per informazioni dettagliate 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 asimmetrica. KMS

L'`DescribeKey`operazione restituisce anche i seguenti dettagli per le chiavi asimmetriche. KMS

- Per le KMS chiavi asimmetriche con un `KeyUsage` valore di`ENCRYPT_DECRYPT`, l'operazione restituisce il`EncryptionAlgorithms`, che elenca gli algoritmi di crittografia validi per la chiave.
- Per le KMS chiavi asimmetriche con un `KeyUsage` valore di`SIGN_VERIFY`, l'operazione restituisce, che elenca gli algoritmi di firma `SigningAlgorithms` validi per la chiave.
- Per le KMS chiavi asimmetriche con un `KeyUsage` valore di`KEY_AGREEMENT`, l'operazione restituisce`KeyAgreementAlgorithms`, che elenca gli algoritmi di accordo chiave validi per la chiave.

Per ulteriori informazioni sulle chiavi asimmetriche, vedereKMS. the section called “Chiavi asimmetriche”

Identifica le chiavi HMAC KMS

Nella AWS KMS console

HMACKMSIe chiavi sono incluse nella tabella Customer managed keys, ma non è possibile ordinare o filtrare questa tabella in base alle specifiche chiave o ai valori di utilizzo delle chiavi che identificano HMAC le chiavi. Per facilitare la ricerca delle HMAC chiavi, assegna 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 KMS chiave mostra il Tipo di chiave, che indica se la chiave è simmetrica o asimmetrica. HMACKMSIe chiavi sono simmetriche. La scheda Configurazione crittografica mostra anche l'utilizzo delle chiavi. Per HMAC KMS le chiavi, il valore di utilizzo della chiave è sempre Genera e verifica MAC.

Nelle DescribeKey risposte

Quando si richiama l'`DescribeKey`operazione su un HMAC KMS tasto, la risposta include i `KeyUsage` valori `KeySpec` and. Per HMAC KMS le chiavi, 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 HMAC KMS chiavi, vederethe section called “HMACchiavi”.

Identifica le chiavi multiregionali KMS

Nella console AWS KMS

La tabella delle chiavi gestite dal cliente mostra solo KMS le chiavi 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 superiore destro 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 KMS chiave”.

La pagina di dettaglio per le KMS chiavi 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, AWS KMS API le operazioni sono regionali e restituiscono solo le risorse nella regione corrente o specificata. Tuttavia, quando si richiama l'`DescribeKey`operazione su una KMS chiave multiregionale, la risposta include tutte le chiavi multiregionali correlate in altre AWS regioni nell'`MultiRegionConfiguration`elemento.

Per ulteriori informazioni sulle KMS chiavi multiregionali, vedere. the section called “Chiavi multi-regione”

Identifica KMS le chiavi con materiale chiave importato

Nella AWS KMS console

Per facilitare l'identificazione KMS delle chiavi con materiale chiave importato nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origin semplifica l'identificazione KMS delle chiavi con un valore della proprietà di origine esterna (materiale chiave di importazione). Per assistenza, consulta the section called “Personalizza le tue tabelle KMS chiave”.

La scheda Configurazione crittografica nella pagina dei dettagli di una KMS chiave mostra l'origine, che identifica la fonte del materiale chiave per la KMS chiave. Per KMS le chiavi 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. La scheda Materiale chiave viene visualizzata solo nella pagina di dettaglio KMS delle chiavi con materiale chiave importato.

Nelle DescribeKey risposte

Quando si richiama l'`DescribeKey`operazione su una KMS chiave con materiale chiave importato`Origin`, la risposta include `ValidTo` i valori`ExpirationModel`, e. Per KMS le chiavi

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. Per ulteriori informazioni, consulta the section called “Impostazione di una data di scadenza (facoltativo)”.

Per ulteriori informazioni sulle KMS chiavi con materiale chiave importato, vedere. the section called “Materiale della chiave importato”

Identifica KMS le chiavi negli archivi AWS CloudHSM delle chiavi

Nella AWS KMS console

Per semplificare l'identificazione KMS delle chiavi negli archivi di AWS CloudHSM chiavi nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origin semplifica l'identificazione KMS delle chiavi con un valore della proprietà AWS CloudHSMorigin. Per assistenza, consulta the section called “Personalizza le tue tabelle KMS chiave”.

La scheda Configurazione crittografica nella pagina dei dettagli di una KMS chiave mostra l'origine, che identifica la fonte del materiale chiave per la KMS chiave. Per KMS le chiavi negli archivi di AWS CloudHSM chiavi, il valore di origine è sempre. AWS CloudHSM

Per una KMS chiave in un archivio di AWS CloudHSM chiavi, la scheda Configurazione crittografica include una sezione aggiuntiva, Archivio chiavi personalizzato, che fornisce informazioni sull'archivio AWS CloudHSM chiavi e sul AWS CloudHSM cluster associati alla KMS chiave.

Nelle risposte DescribeKey

Quando si richiama l'`DescribeKey`operazione su una KMS chiave in un archivio di AWS CloudHSM chiavi, la risposta include il`Origin`, che identifica l'origine del materiale chiave. Per KMS le chiavi in un archivio di AWS CloudHSM chiavi, il valore di origine è sempre`AWS_CLOUDHSM`. L'operazione restituisce anche i seguenti campi speciali per KMS le chiavi negli archivi di AWS CloudHSM chiavi:

- `CloudHsmClusterId`
- `CustomKeyStoreId`

Per ulteriori informazioni sugli archivi di AWS CloudHSM chiavi, vederethe section called “AWS CloudHSM negozi chiave”.

Identifica KMS le chiavi negli archivi di chiavi esterni

Nella AWS KMS console

Per semplificare l'identificazione KMS delle chiavi negli archivi di chiavi esterni nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origin semplifica l'identificazione KMS delle chiavi con un valore della proprietà External key store origin. Per assistenza, consulta the section called “Personalizza le tue tabelle KMS chiave”.

La scheda Configurazione crittografica nella pagina dei dettagli di una KMS chiave mostra l'origine, che identifica la fonte del materiale chiave per la KMS chiave. Per KMS le chiavi negli archivi di chiavi esterni, il valore di origine è sempre Archivio chiavi esterno.

Per una KMS chiave 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 KMS chiave. La tabella delle chiavi esterne viene visualizzata nella AWS KMS console solo per KMS le chiavi negli archivi di chiavi esterni. Fornisce informazioni sulla chiave esterna associata alla KMS chiave. La chiave esterna è una chiave crittografica esterna AWS che funge da materiale chiave per la KMS chiave nell'archivio delle chiavi esterno. Quando si esegue la crittografia o la decrittografia con la KMS chiave, l'operazione viene eseguita dal gestore delle chiavi esterno 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. L'ID della chiave esterna viene specificato al momento della creazione della KMS chiave e non è possibile modificarlo. Se il valore dell'ID della chiave esterna utilizzato per creare la KMS chiave cambia o diventa non valido, è necessario pianificare l'eliminazione della KMS chiave e creare una nuova KMS chiave con il valore ID della chiave esterna corretto.

Nelle risposte DescribeKey

Quando si richiama l'`DescribeKey`operazione su una KMS chiave in un archivio di chiavi esterno, la risposta include il`Origin`, che identifica l'origine del materiale chiave. Per KMS le chiavi in un archivio di AWS CloudHSM 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 KMS chiavi.

Per ulteriori informazioni sugli archivi di chiavi esterni, vederethe section called “Archivi delle chiavi esterne”.

Personalizza la visualizzazione della console

Puoi personalizzare la visualizzazione della AWS KMS console per facilitare la ricerca KMS delle chiavi. Personalizza le tabelle visualizzate nelle pagine Chiavi gestite da AWSe nelle pagine relative alle chiavi gestite dai clienti per visualizzare le informazioni di cui hai più bisogno oppure ordina e filtra le KMS chiavi restituite nelle tabelle.

Argomenti

Ordina e filtra le tue KMS chiavi

Per facilitare la ricerca KMS delle chiavi nella console, puoi ordinare e filtrare le tabelle delle chiavi.

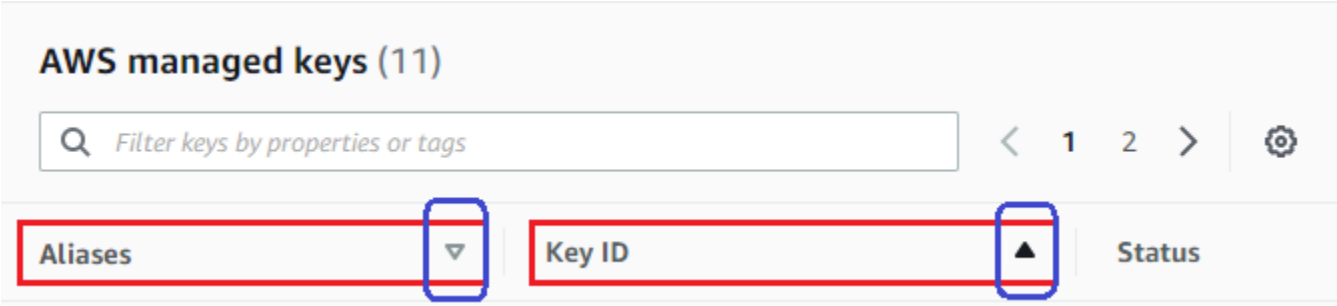
Ordina

È possibile ordinare KMS le chiavi in ordine crescente o decrescente in base ai valori delle colonne. Questa funzione ordina tutte le KMS chiavi della tabella, anche se non compaiono 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, è possibile ordinare KMS le chiavi in ordine crescente in base all'ID della chiave, anziché agli alias, che è l'impostazione predefinita.



Quando si ordinano KMS le chiavi nella pagina Chiavi gestite dal cliente in ordine crescente per tipo di chiave, tutte le chiavi asimmetriche vengono visualizzate prima di tutte le chiavi simmetriche.

Filtro

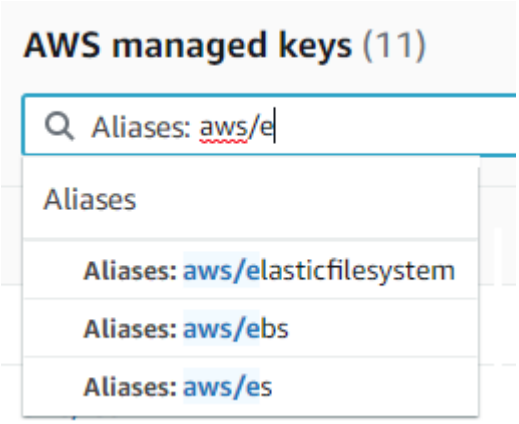
È possibile filtrare le KMS chiavi in base ai valori delle proprietà o ai tag. Il filtro si applica a tutte le KMS chiavi della tabella, anche se non compaiono nella pagina della tabella corrente. Il filtro non fa distinzione tra maiuscole e minuscole.

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 KMS le chiavi con un nome alias che contiene`aws/e`, scegliete la casella del filtro, scegliete Alias`aws/e`, digitate e quindi premete `Enter` o `Return` per aggiungere il filtro.



Filtri di tabella KMS chiave consigliati

Filtro per chiavi asimmetriche KMS

Per visualizzare solo KMS le chiavi asimmetriche nella pagina Chiavi gestite dal cliente, fai clic sulla casella del filtro, scegli Tipo di chiave, quindi scegli Tipo di chiave: asimmetrico. L'opzione Asimmetrica viene visualizzata solo quando nella tabella sono presenti chiavi asimmetriche. KMS

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 KMS le chiavi con un tag particolare, scegliete la casella del filtro, scegliete la chiave del tag, quindi scegliete uno dei valori effettivi del tag. È anche possibile digitare o tutto il valore del tag o solo parte di esso.

La tabella risultante mostra tutte le KMS chiavi con il tag scelto. Tuttavia, il tag non viene visualizzato. Per visualizzare il tag, scegli l'ID o l'alias della KMS chiave e nella relativa pagina di dettaglio, 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 KMS le chiavi digitando solo la chiave del tag o solo il suo valore. Per filtrare i tag in base alla chiave o al valore del tag, usa l'ListResourceTagsoperazione per ottenere KMS le chiavi 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 KMS le chiavi con `test` tag chiavi o proprietà filtrabili, digitate `test` nella casella del filtro. L'anteprima mostra le KMS chiavi che verranno selezionate dal filtro. In questo caso, `test` appare solo nella proprietà Alias.

Personalizza le tue tabelle KMS chiave

È possibile personalizzare le tabelle visualizzate nelle pagine relative alle chiavi gestite dal cliente Chiavi gestite da AWSe quelle relative alle chiavi gestite dal AWS Management Console cliente in base alle proprie 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 KMS principali

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).

Valuta la possibilità di utilizzare l'impostazione delle dimensioni della pagina per aumentare il numero di KMS tasti visualizzati su ogni pagina, soprattutto se in genere utilizzi un dispositivo facile da scorrere.

Le colonne di dati visualizzate possono variare a seconda della tabella, del ruolo professionale e dei tipi di KMS chiavi presenti 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 KMS delle tabelle chiave consigliate

È possibile personalizzare le colonne visualizzate nella tabella KMS delle chiavi per visualizzare le informazioni necessarie sulle KMS chiavi.

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 di crittografia simmetriche KMS

Se si utilizzano solo KMS chiavi di crittografia simmetriche con materiale chiave generato da AWS KMS, è probabile che le colonne Alias, Key ID, Status e Creation date siano le più utili.

Chiavi asimmetriche KMS

Se utilizzi KMS chiavi asimmetriche, oltre alle colonne Alias, Key ID e Status, valuta la possibilità di aggiungere le colonne Key type, Key spec e Key usage. Queste colonne mostreranno se una KMS chiave è simmetrica o asimmetrica, il tipo di materiale della chiave e se la chiave può essere utilizzata per la crittografia o la firma. KMS

HMACKMSchiavi

Se utilizzi HMAC KMS le chiavi, oltre alle colonne Alias, Key ID e Status, valuta la possibilità di aggiungere le colonne Key spec e Key usage. Queste colonne ti mostreranno se una KMS chiave è una HMAC chiave. Poiché non puoi ordinare KMS le chiavi in base alle specifiche o all'utilizzo della chiave, utilizza alias e tag per identificare le HMAC chiavi e quindi utilizza le funzionalità di filtro della AWS KMS console per filtrare in base a alias o tag.

Materiale della chiave importato

Se disponi di KMS chiavi con materiale chiave importato, valuta la possibilità di aggiungere le colonne Origine e Data di scadenza. Queste colonne mostreranno se il materiale chiave contenuto in una KMS chiave 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 KMS chiave (senza materiale chiave). Non riflette alcuna caratteristica del materiale della chiave.

Chiavi nello store delle chiavi personalizzate

Se disponi di KMS chiavi negli archivi chiavi personalizzati, valuta la possibilità di aggiungere le colonne Origin e Custom key store ID. Queste colonne mostrano che la KMS chiave si trova in

un archivio chiavi personalizzato, mostrano il tipo di archivio chiavi personalizzato e identificano l'archivio chiavi personalizzato.

Chiavi multi-regione

Se hai chiavi multi-regione, prendi in considerazione l'aggiunta della colonna Regionalità. Ciò mostra se una KMS chiave è una chiave a regione singola, una chiave primaria multiregionale o una chiave di replica multiregionale.

Trova KMS chiavi e materiale chiave in un negozio di AWS CloudHSM chiavi

Se gestisci un archivio di AWS CloudHSM chiavi, potrebbe essere necessario identificare le KMS chiavi in ogni archivio di AWS CloudHSM chiavi. Ad esempio, potresti aver bisogno di eseguire alcune delle operazioni seguenti.

- Tieni traccia delle KMS chiavi nell'archivio AWS CloudHSM chiavi nei AWS CloudTrail registri.
- Prevedi l'effetto sui KMS tasti della disconnessione di un archivio di AWS CloudHSM chiavi.
- Pianifica l'eliminazione delle KMS chiavi prima di eliminare un archivio di AWS CloudHSM chiavi.

Inoltre, potresti voler identificare le chiavi del tuo AWS CloudHSM cluster che fungono da materiale chiave per KMS le tue chiavi. Sebbene AWS KMS gestisca le KMS chiavi e il materiale chiave, l'utente mantiene comunque il controllo e la responsabilità della gestione del AWS CloudHSM cluster, nonché dei backup HSMs e delle chiavi delHSMs. 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 KMS chiavi nel tuo archivio delle AWS CloudHSM chiavi 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 KMS chiave.

Per trovare KMS chiavi e materiale chiave, usa una delle seguenti tecniche.

- Trova le KMS chiavi in un archivio di AWS CloudHSM chiavi— Come identificare le KMS chiavi in uno o tutti i tuoi AWS CloudHSM portachiavi.
- Trova tutte le chiavi di un archivio di AWS CloudHSM chiavi— Come trovare tutte le chiavi nel cluster che fungono da materiale chiave per le KMS chiavi nell'archivio AWS CloudHSM delle chiavi.

- Trova la AWS CloudHSM chiave per una KMS chiave— Come trovare la chiave nel cluster che funge da materiale chiave per una particolare KMS chiave nel proprio archivio di AWS CloudHSM chiavi.
- Trova la KMS chiave per una AWS CloudHSM chiave— Come trovare la KMS chiave per una particolare chiave nel cluster.

Trova le KMS chiavi in un archivio di AWS CloudHSM chiavi

Se gestisci un archivio di AWS CloudHSM chiavi, potrebbe essere necessario identificare le KMS chiavi in ogni archivio di AWS CloudHSM chiavi. È possibile utilizzare queste informazioni per tenere traccia delle operazioni KMS chiave nei AWS CloudTrail registri, prevedere l'effetto della disconnessione di un archivio chiavi personalizzato sulle KMS chiavi o pianificare l'eliminazione delle KMS chiavi prima di eliminare un archivio AWS CloudHSM chiavi.

Per trovare le KMS chiavi in un archivio di AWS CloudHSM chiavi (console)

Per trovare le KMS chiavi 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 KMS le chiavi in qualsiasi archivio di AWS CloudHSM chiavi, cerca KMS le chiavi 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 KMS chiavi in un archivio di AWS CloudHSM chiavi (API)

Per trovare le KMS chiavi in un archivio di AWS CloudHSM chiavi, usa le DescribeKeyoperazioni ListKeysand, quindi filtra per `CustomKeyStoreId` valore. Prima di eseguire gli esempi seguenti, sostituite i valori ID fittizi del key store personalizzato con un valore valido.

Bash

Per trovare KMS le chiavi in un determinato archivio AWS CloudHSM chiavi, recupera tutte le KMS chiavi presenti 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 '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

Per ottenere KMS le chiavi in qualsiasi AWS CloudHSM archivio di chiavi dell'account e della regione, cerca `CustomKeyStoreType` con un 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 '"CustomKeyStoreType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

Per trovare KMS le chiavi in un particolare archivio di AWS CloudHSM chiavi, utilizzare i KmsKey cmdlet Get - KmsKeyList e Get - per ottenere tutte le KMS chiavi 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 KMS le chiavi in qualsiasi archivio di AWS CloudHSM chiavi dell'account e della regione, filtra in base al CustomKeyStoreType valore di. `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. Per farlo, usa il comando key list in Cloud HSMCLI.

Puoi anche usare il comando key list per trovare la AWS CloudHSM chiave AWS KMS for an. Quando AWS KMS crea il materiale chiave per una KMS chiave nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della KMS chiave nell'etichetta della chiave. Il comando key list restituisce `key-reference` e il`label`.

> **Note**
>
> Le seguenti procedure utilizzano lo strumento da riga di comando di AWS CloudHSM Client SDK 5, Cloud HSM CLI. Il Cloud HSM CLI sostituisce `key-handle` con`key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

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 chiavi personalizzato è disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare le KMS chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Usa il comando key list in Cloud HSM CLI per trovare tutte le chiavi per l'utente corrente presente nel tuo AWS CloudHSM cluster.

   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 AWS CloudHSM per l'utente.

```
aws-cloudhsm > key list
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000000123",
        "attributes": {
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
      },
      {
        "key-reference": "0x0000000000000456",
```

```
        "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 KMS chiave per una AWS CloudHSM chiave

Se conosci il riferimento o l'ID di una chiave di cui è `kmsuser` proprietaria nel cluster, puoi utilizzare quel valore per identificare la KMS chiave associata nel tuo archivio di AWS CloudHSM chiavi.

Quando AWS KMS crea il materiale chiave per una KMS chiave nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della KMS chiave nell'etichetta della chiave. A meno che tu non abbia modificato il valore dell'etichetta, puoi utilizzare il comando [key list](#) in Cloud HSM CLI per identificare la KMS chiave associata alla AWS CloudHSM chiave.

> **Note**
>
> Le seguenti procedure utilizzano lo strumento da riga di comando di AWS CloudHSM Client SDK 5, [Cloud HSM CLI](#). Il Cloud HSM CLI sostituisce `key-handle` con`key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida](#) per l'AWS CloudHSM utente.

Per eseguire queste procedure è necessario disconnettere temporaneamente l'archivio delle AWS CloudHSM chiavi in modo da poter accedere come CU. `kmsuser`

> **Note**
>
> Mentre un archivio chiavi personalizzato è disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare le KMS chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Argomenti

Identifica la KMS chiave associata a un riferimento chiave

Le seguenti procedure mostrano come utilizzare il comando key list in Cloud HSM CLI con il filtro `key-reference` degli attributi per trovare la chiave nel cluster che funge da materiale chiave per una particolare KMS chiave nel tuo archivio di AWS CloudHSM chiavi.

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi accedi come `kmsuser` spiegato in. Come disconnettersi ed eseguire l'accesso
2. Usa il comando key list in Cloud HSM CLI per filtrare in base all'`key-reference`attributo. Specificate l'`verbose`argomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'`verbose`argomento, 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 inCome scollegarsi e riconnettersi.

Identifica la KMS chiave associata all'ID di una chiave di backup

Tutte le voci di CloudTrail registro per le operazioni crittografiche con una KMS chiave in un archivio di AWS CloudHSM chiavi includono un `additionalEventData` campo con e. `customKeyStoreId` `backingKeyId` Il valore restituito nel `backingKeyId` campo è correlato all'attributo HSM chiave `id` Cloud. È possibile filtrare l'operazione dell'elenco di chiavi in base all'`id`attributo per identificare la KMS chiave associata a uno specifico`backingKeyId`.

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi accedi come `kmsuser` spiegato in. Come disconnettersi ed eseguire l'accesso
2. Usa il comando key list in Cloud HSM CLI con il filtro degli attributi per trovare la chiave nel tuo cluster che funge da materiale chiave per una particolare KMS chiave nel tuo archivio di AWS CloudHSM chiavi.

   L'esempio seguente mostra come filtrare in base all'`id`attributo. 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`

      ```
      0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138
      ```

   c. Utilizzate il `backingKeyId` valore convertito per filtrare in base all'attributo. `id` Specificate l'`verbose`argomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'`verbose`argomento, 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": "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":
 "0x353839303732333632326463313566363939616139616232333837613966373434623262383834633133
          "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 KMS chiave

È possibile utilizzare l'KMSID di una KMS chiave 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 KMS chiave nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della KMS chiave nell'etichetta della chiave. A meno che tu non abbia modificato il valore dell'etichetta, puoi utilizzare il comando key list in Cloud HSM CLI per trovare la risorsa chiave e l'id del materiale chiave per la KMS chiave.

Tutte le voci di CloudTrail registro per le operazioni crittografiche con una KMS chiave in un archivio di AWS CloudHSM chiavi includono un `additionalEventData` campo con la e. `customKeyStoreId` `backingKeyId` Il valore restituito nel `backingKeyId` campo è l'attributo `id` AWS CloudHSM chiave. È possibile filtrare l' AWS CloudHSM CLIoperazione dell'elenco di KMS chiavi ARN per chiave per identificare l'`id`attributo HSM chiave Cloud associato a una KMS chiave specifica.

Per eseguire questa procedura, è necessario disconnettere temporaneamente l'archivio delle AWS CloudHSM chiavi in modo da poter accedere come `kmsuser` CU.

> **Note**
>
> Le seguenti procedure utilizzano lo strumento da riga di comando di AWS CloudHSM Client SDK 5, Cloud. HSM CLI Il Cloud HSM CLI sostituisce `key-handle` con`key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

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

   > **Note**
   >
   > Quando un archivio chiavi personalizzato è disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare le KMS chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Usa il comando key list in Cloud HSM CLI e filtra `label` per trovare la KMS chiave per una particolare chiave nel tuo AWS CloudHSM cluster. Specificate l'`verbose`argomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'`verbose`argomento, 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 KMS chiave 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": "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](#).

Attivazione e disattivazione dei tasti

È possibile disabilitare e riabilitare le chiavi gestite del cliente. Quando si crea una KMS chiave, questa è abilitata per impostazione predefinita. Se si disabilita una KMS chiave, questa non può essere utilizzata in alcuna operazione di crittografia finché non viene riattivata.

Poiché è temporanea e facilmente annullabile, la disattivazione di una KMS chiave è un'alternativa sicura all'eliminazione di una KMS chiave, un'azione distruttiva e irreversibile. Se stai pensando di eliminare una KMS chiave, disattivala prima e imposta un CloudWatch allarme o un meccanismo simile per essere certo di non dover mai usare la chiave per decrittografare i dati crittografati.

Quando disabiliti una KMS chiave, questa diventa immediatamente inutilizzabile (fatta salva l'eventuale coerenza). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono influenzate fino a quando la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

Non è possibile abilitare o disabilitare Chiavi gestite da AWSo 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 AWS Management Console 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. Scegliete la casella di controllo relativa ai KMS tasti che desiderate attivare o disattivare.
5. Per abilitare una KMS chiave, scegli Azioni chiave, Abilita. Per disabilitare una KMS chiave, scegli Azioni chiave, Disabilita.

Usando il AWS KMS API

L'EnableKeyoperazione abilita un disabile 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'DescribeKeyoperazione.

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

L'DisableKeyoperazione disabilita una KMS chiave 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'DescribeKeyoperazione 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 dai clienti, potete creare nuove KMS chiavi e quindi modificare le applicazioni o gli alias per utilizzare le nuove chiavi. KMS In alternativa, è possibile ruotare il materiale chiave associato a una chiave esistente abilitando KMS la rotazione automatica delle chiavi o eseguendo la rotazione su richiesta.

Per impostazione predefinita, quando abiliti la rotazione automatica delle chiavi per una KMS chiave, AWS KMS genera nuovo materiale crittografico per la KMS chiave ogni anno. È inoltre possibile specificare un'impostazione personalizzata rotation-period per definire il numero di giorni dopo l'attivazione della rotazione automatica dei tasti 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.

AWS KMS salva tutte le versioni precedenti del materiale crittografico in modo perpetuo in modo da poter decrittografare tutti i dati crittografati con quella chiave. KMS AWS KMS non elimina alcun materiale relativo alla chiave ruotata finché l'utente non elimina la chiave. KMS Puoi monitorare la rotazione del materiale chiave per KMS le tue chiavi in Amazon CloudWatch e nella AWS Key Management Service console. AWS CloudTrail Puoi anche utilizzare l'GetKeyRotationStatusoperazione per verificare se la rotazione automatica è abilitata per una KMS chiave e identificare eventuali rotazioni in corso su richiesta. È possibile utilizzare l'ListKeyRotationsoperazione per visualizzare i dettagli delle rotazioni completate.

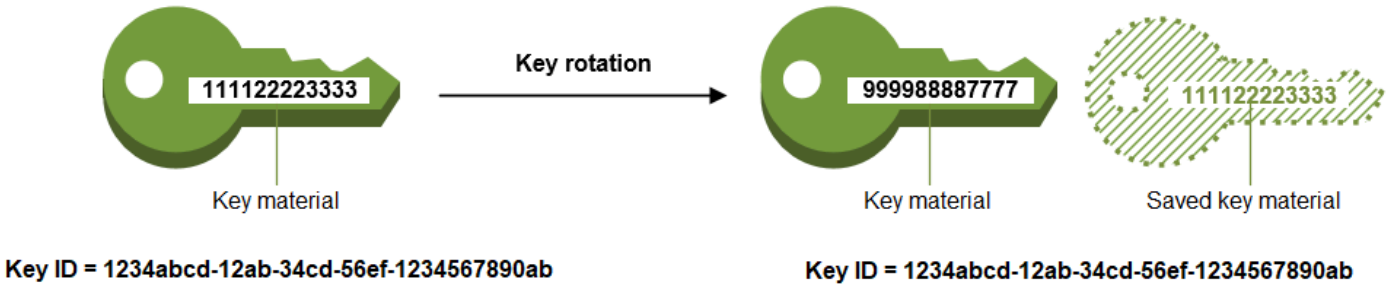
Quando si utilizza una KMS chiave ruotata per crittografare i dati, AWS KMS utilizza il materiale chiave corrente. Quando si utilizza la KMS chiave ruotata per decrittografare il testo cifrato, AWS KMS utilizza la versione del materiale chiave utilizzato per crittografarlo. Non è possibile selezionare una versione particolare del materiale chiave per le operazioni di decrittografia, sceglie automaticamente la versione corretta. AWS KMS Poiché esegue la decrittografia AWS KMS in modo trasparente con il materiale chiave appropriato, è possibile utilizzare in sicurezza una chiave ruotata nelle applicazioni e senza modifiche al KMS codice. Servizi AWS

Tuttavia, la rotazione automatica delle chiavi non ha alcun effetto sui dati che la chiave protegge. KMS Non ruota le chiavi dati generate dalla KMS chiave né cripta nuovamente i dati protetti dalla KMS chiave e non mitiga l'effetto di una chiave dati compromessa.

AWS KMS supporta la rotazione automatica e su richiesta delle chiavi solo per le chiavi di crittografia simmetriche con il materiale chiave creato. KMS AWS KMS La rotazione automatica è facoltativa per le chiavi gestite dal cliente. KMS AWS KMS ruota sempre ogni anno il materiale chiave per KMSle chiavi AWS gestite. La rotazione delle KMSchiavi di AWS proprietà è gestita dal AWS servizio proprietario della chiave.

> **Note**
>
> Il periodo di rotazione Chiavi gestite da AWS è cambiato a maggio 2022. Per informazioni dettagliate, consultare Chiavi gestite da AWS.

La rotazione della chiave modifica solo il materiale della chiave, cioè il segreto crittografico utilizzato nelle operazioni di crittografia. La KMS chiave è la stessa risorsa logica, indipendentemente dal fatto che il materiale chiave venga modificato o meno. Le proprietà della KMS chiave non cambiano, come mostrato nell'immagine seguente.



Potresti decidere di creare una nuova KMS chiave e usarla al posto della KMS chiave originale. Ciò ha lo stesso effetto della rotazione del materiale chiave in una KMS chiave esistente, quindi spesso si pensa che si tratti di una rotazione manuale della chiave. La rotazione manuale è un'ottima scelta quando si desidera ruotare KMS chiavi che non sono idonee alla rotazione automatica, tra cui chiavi asimmetriche, KMS chiavi, chiavi negli archivi HMACKMSchiavi personalizzati e KMS KMS chiavi con materiale chiave importato.

Rotazione delle chiave 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 visualizzazione per visualizzare i costi totali per le chiavi fatturate come chiavi correnti e ruotate KMS specificando il Tipo di utilizzo e raggruppando i dati **`$REGION-KMS-Keys`** per Operazione. API
> Potresti ancora visualizzare le istanze dell'operazione precedente `Unknown` API per le date storiche.

Rotazione delle chiavi e quote

Ogni KMS chiave conta come una chiave nel calcolo delle quote di risorse chiave, indipendentemente dal numero di versioni del materiale a chiave ruotata.

Per informazioni più dettagliate sul materiale chiave e sulla rotazione, consultare Dettagli di crittografia di AWS Key Management Service.

Argomenti

- Perché ruotare i tasti? KMS
- Come funziona la rotazione dei tasti
- Abilita la rotazione automatica dei tasti
- Disabilita la rotazione automatica dei tasti
- Esegue la rotazione dei tasti su richiesta
- Ruota i tasti manualmente
- Cambia la chiave primaria in un set di chiavi multiregionali

Perché ruotare i tasti? 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. Per evitare questo esaurimento delle chiavi, è preferibile utilizzare le chiavi di dati solo una volta o poche volte, in modo da ruotare efficacemente il materiale delle chiavi.

Tuttavia, KMS le chiavi vengono spesso utilizzate come chiavi di avvolgimento, note anche come chiavi di crittografia a chiave. 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.

Nonostante questo rischio di esaurimento molto basso, potrebbe essere necessario ruotare KMS le chiavi a causa di regole aziendali o contrattuali o normative governative. Quando si è costretti a ruotare i KMS tasti, si consiglia di utilizzare la rotazione automatica dei tasti laddove è supportata, e la rotazione manuale dei tasti quando la rotazione automatica dei tasti non è supportata.

Potresti prendere in considerazione l'idea di eseguire 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

La rotazione dei tasti AWS KMS 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 KMS chiave alla data di rotazione successiva definita dal periodo di rotazione. Non devi ricordare o pianificare l'aggiornamento.

Rotazione su richiesta

Avvia immediatamente la rotazione del materiale chiave associato alla KMS chiave, indipendentemente dal fatto che la rotazione automatica della chiave sia abilitata o meno.

Gestione del materiale chiave

AWS KMS mantiene tutto il materiale chiave di una KMS chiave, anche se la rotazione dei tasti è disabilitata. AWS KMS elimina il materiale chiave solo quando si elimina la KMS chiave.

Utilizzo del materiale chiave

Quando si utilizza una KMS chiave ruotata per crittografare i dati, AWS KMS utilizza il materiale chiave corrente. Quando si utilizza la KMS chiave ruotata per decrittografare il testo cifrato, AWS

KMS utilizza la stessa versione del materiale chiave utilizzato per crittografarlo. Non è possibile selezionare una versione particolare del materiale chiave per le operazioni di decrittografia, sceglie automaticamente la versione corretta. AWS KMS

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

AWS KMS ruota automaticamente la KMS chiave 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 modifichi il periodo di rotazione per una chiave su cui in precedenza avevi abilitato la rotazione automatica dei tasti. 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.

- Disattiva la rotazione dei tasti: se disattivate la rotazione automatica dei tasti in qualsiasi momento, la KMS chiave continua a utilizzare la versione del materiale 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.

- KMSTasti disattivati: mentre un KMS tasto è disabilitato, AWS KMS non lo ruota. Tuttavia, lo stato di rotazione dei tasti non cambia e non è possibile modificarlo mentre il KMS tasto è disabilitato. Quando la KMS chiave viene riattivata, se il materiale della chiave ha superato l'ultima data di rotazione programmata, AWS KMS la ruota immediatamente. Se il materiale chiave non ha perso l'ultima data di rotazione programmata, AWS KMS riprende la pianificazione di rotazione della chiave originale.
- KMSchiavi in attesa di cancellazione: mentre una KMS chiave è in attesa di cancellazione, AWS KMS non la ruota. 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 AWS KMS ruota immediatamente. 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 gestite da AWS I nuovi vengono ruotati automaticamente un anno dopo la creazione e successivamente all'incirca ogni anno.
> Chiavi gestite da AWS Le versioni esistenti vengono ruotate automaticamente un anno dopo la loro rotazione più recente e successivamente ogni anno.

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 an 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.

Tipi di KMS chiave supportati

La rotazione automatica delle chiavi è supportata solo sulle KMSchiavi di crittografia simmetriche con materiale chiave AWS KMS generato (Origin =AWS_KMS).

La rotazione automatica delle chiavi non è supportata sui seguenti tipi di KMS chiavi, ma è possibile ruotarle manualmente. KMS

- Tasti asimmetrici KMS
- HMACKMSchiavi
- KMSchiavi negli archivi di chiavi personalizzati
- KMSchiavi con materiale chiave importato

Tasti rotanti per più regioni

È possibile abilitare e disabilitare la rotazione automatica ed eseguire la rotazione su richiesta del materiale chiave nelle chiavi multiregionali con crittografia simmetrica. 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

Puoi abilitare la rotazione automatica delle chiavi per le chiavi gestite dal cliente utilizzate per la crittografia lato server nei servizi AWS . La rotazione annuale è trasparente e compatibile con i servizi AWS .

Monitoraggio della rotazione delle chiavi

Quando AWS KMS ruota il materiale chiave per una chiave Chiave gestita da AWSo 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 KMS chiave sia stata ruotata.

È possibile 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'ListKeyRotationsoperazione 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 della KMS chiave, AWS KMS genera nuovo materiale crittografico per la KMS chiave ogni anno. È inoltre possibile specificare un'impostazione personalizzata rotation-period per definire il numero di giorni dopo l'attivazione della rotazione automatica dei tasti 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 KMS chiave, inclusi l'ID, la chiave, la regioneARN, le politiche e le autorizzazioni, non cambiano quando la chiave viene ruotata.
- Non è necessario modificare le applicazioni o gli alias che si riferiscono all'ID o alla chiave ARN della chiave. KMS
- Il materiale rotante della chiave non influisce in alcun modo sull'uso della KMS chiave. Servizio AWS
- Dopo aver abilitato la rotazione dei tasti, AWS KMS ruota automaticamente il KMS tasto 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'EnableKeyRotationoperazione. 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 AWS Management Console 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. Scegliete l'alias o l'ID della chiave. KMS
5. Scegliere la scheda Key rotation (Rotazione chiave).

   La scheda Rotazione delle chiavi viene visualizzata solo nella pagina di dettaglio delle KMS chiavi di crittografia simmetriche con il materiale chiave AWS KMS generato (l'origine è AWS_KMS), incluse le chiavi di crittografia simmetriche multiregione. KMS

   Non è possibile ruotare automaticamente chiavi asimmetriche, KMS chiavi, chiavi con materiale chiave importato o HMAC KMS KMS chiavi negli archivi di chiavi personalizzati. KMS Tuttavia è possibile ruotare queste chiavi manualmente.
6. Nella sezione Rotazione automatica dei tasti, scegliete Modifica.
7. Per Rotazione dei tasti, selezionate Abilita.

> **Note**
>
> Se una KMS chiave 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 dei tasti. Abilita la KMS chiave 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. Seleziona Salva.

Usando il AWS KMS API

È possibile utilizzare AWS Key Management Service (AWS KMS) API per abilitare la rotazione automatica dei tasti 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'EnableKeyRotationoperazione consente la rotazione automatica dei tasti per la KMS chiave specificata. Per identificare la KMS chiave in questa operazione, utilizzate l'ID o la chiave corrispondente ARN. Per impostazione predefinita, la rotazione automatica è disabilitata per le chiavi gestite dal cliente.

È possibile utilizzare la chiave di kms:RotationPeriodInDayscondizione per limitare i valori che i principali possono specificare per il `RotationPeriodInDays` parametro di una `EnableKeyRotation` richiesta.

L'esempio seguente abilita la rotazione delle chiavi con un periodo di rotazione di 180 giorni sulla KMS chiave di crittografia simmetrica specificata e utilizza l'GetKeyRotationStatusoperazione 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 dei tasti

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 KMS chiave continua a utilizzare la versione del materiale 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.

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 AWS Management Console 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. Scegliete l'alias o l'ID della chiave. KMS
5. Scegliere la scheda Key rotation (Rotazione chiave).

La scheda Rotazione delle chiavi viene visualizzata solo nella pagina di dettaglio delle KMS chiavi di crittografia simmetriche con il materiale chiave AWS KMS generato (l'origine è AWS_KMS), incluse le chiavi di crittografia simmetriche multiregione. KMS

Non è possibile ruotare automaticamente chiavi asimmetriche, KMS chiavi, chiavi con materiale chiave importato o HMAC KMS KMS chiavi negli archivi di chiavi personalizzati. KMS Tuttavia è possibile ruotare queste chiavi manualmente.

6. Nella sezione Rotazione automatica dei tasti, scegliete Modifica.
7. Per Rotazione dei tasti, selezionate Disabilita.

> **Note**
>
> Se una KMS chiave 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 dei tasti. Abilita la KMS chiave 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. Seleziona Salva.

Usando il AWS KMS API

È possibile utilizzare AWS Key Management Service (AWS KMS) API per disabilitare la rotazione automatica dei tasti 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'DisableKeyRotationoperazione disabilita la rotazione automatica dei tasti. Per identificare la KMS chiave in questa operazione, utilizzate l'ID o la chiave corrispondente. 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 di crittografia KMS simmetrica specificata e utilizza l'GetKeyRotationStatusoperazione per visualizzare il 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
}
```

Esegue la rotazione dei tasti su richiesta

È possibile eseguire la rotazione su richiesta del materiale chiave nelle KMS chiavi 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, si consideri una KMS chiave con la 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 È possibile 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 di crittografia simmetriche. KMS Non è possibile eseguire la rotazione su richiesta di chiavi asimmetriche, KMS chiavi, HMACKMSchiavi con materiale chiave importato o KMS KMS chiavi in un archivio di chiavi personalizzato. Per eseguire la rotazione su richiesta di un set di chiavi multiregionali correlate, richiamate la rotazione su richiesta sulla chiave primaria.

Gli utenti autorizzati possono utilizzare la AWS KMS console e avviare la rotazione dei AWS KMS API tasti su richiesta e visualizzare lo stato di rotazione dei tasti.

Argomenti

- Avvio della rotazione dei tasti su richiesta (console)
- Avvio della rotazione dei tasti su richiesta ()AWS KMS API

Avvio della rotazione dei tasti su richiesta (console)

1. Accedi a AWS Management Console 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 eseguire la rotazione su richiesta di. Chiavi gestite da AWS Vengono ruotate automaticamente ogni anno.)
4. Scegli l'alias o l'ID della chiave. KMS
5. Scegliere la scheda Key rotation (Rotazione chiave).

   La scheda Rotazione delle chiavi viene visualizzata solo nella pagina di dettaglio delle KMS chiavi di crittografia simmetriche con il materiale chiave AWS KMS generato (l'origine è AWS_KMS), incluse le chiavi di crittografia simmetriche multiregione. KMS

   Non è possibile eseguire la rotazione su richiesta di chiavi asimmetriche, KMS chiavi, chiavi con materiale chiave importato o HMAC KMS KMS chiavi negli archivi di chiavi personalizzati. KMS Tuttavia è possibile ruotare queste chiavi manualmente.
6. Nella sezione Rotazione dei tasti su richiesta, scegli la chiave di rotazione.
7. Leggi e considera l'avviso e le informazioni sul numero di rotazioni su richiesta rimanenti della chiave. 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 dei tasti su richiesta ()AWS KMS API

È possibile utilizzare AWS Key Management Service (AWS KMS) API 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'RotateKeyOnDemandoperazione avvia immediatamente la rotazione dei tasti su richiesta per la chiave specificataKMS. Per identificare la KMS chiave in queste operazioni, utilizzate il relativo ID o la chiave. ARN

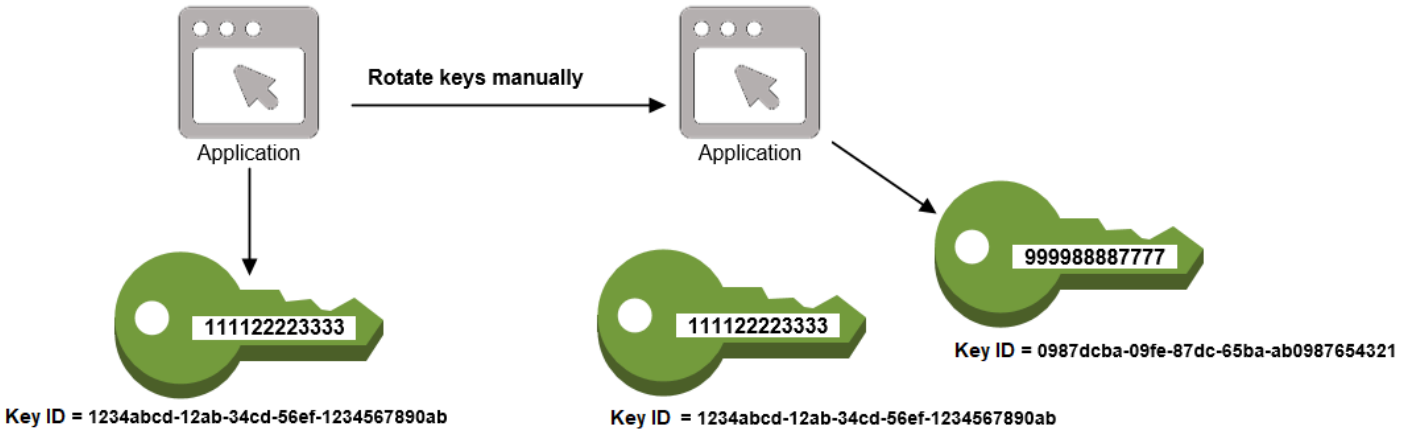
L'esempio seguente avvia la rotazione delle chiavi su richiesta sulla chiave di crittografia KMS simmetrica specificata e utilizza l'GetKeyRotationStatusoperazione per verificare che la rotazione su richiesta sia in corso. `OnDemandRotationStartDate`Nella `kms:GetKeyRotationStatus` risposta identifica la data e l'ora in cui è stata avviata una rotazione su richiesta in corso.

```
$ 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
}
```

Ruota i tasti manualmente

Potresti voler creare una nuova KMS chiave e usarla al posto di una KMS chiave corrente invece di abilitare la rotazione automatica dei tasti. Quando la nuova KMS chiave ha un materiale crittografico diverso rispetto alla KMS chiave corrente, l'utilizzo della nuova KMS chiave ha lo stesso effetto della modifica del materiale della chiave in una KMS chiave esistente. Il processo di sostituzione di una KMS chiave con un'altra è noto come rotazione manuale dei tasti.



La rotazione manuale è un'ottima scelta quando si desidera ruotare KMS chiavi che non sono idonee alla rotazione automatica delle chiavi, come chiavi asimmetriche, KMS chiavi, HMAC KMS chiavi negli archivi KMS chiavi personalizzati e KMS chiavi con materiale chiave importato.

> **Note**
>
> Quando inizi a utilizzare la nuova KMS chiave, assicurati di mantenere attiva la KMS chiave originale in modo da AWS KMS poter decrittografare i dati crittografati dalla chiave originale. KMS

Quando ruotate KMS le chiavi manualmente, dovete aggiornare anche i riferimenti all'ID o alla KMS chiave ARN nelle applicazioni. Gli alias, che associano un nome descrittivo a una KMS chiave, possono semplificare questo processo. Utilizzate un alias per fare riferimento a una KMS chiave nelle vostre applicazioni. Quindi, se desideri modificare la KMS chiave utilizzata dall'applicazione, anziché modificare il codice dell'applicazione, modifica la KMS chiave di destinazione dell'alias. Per informazioni dettagliate, consultare Scopri come utilizzare gli alias nelle tue applicazioni.

> **Note**
>
> Gli alias che rimandano alla versione più recente di una KMS chiave ruotata manualmente sono una buona soluzione per le operazioni DescribeKeyEncrypt, GenerateDataKeyGenerateDataKeyPairGenerateMac, e Sign. Gli alias non sono consentiti nelle operazioni che gestiscono le KMS chiavi, come o. DisableKeyScheduleKeyDeletion Quando si richiama l'operazione Decrypt su KMS chiavi di crittografia simmetriche ruotate manualmente, omettete il parametro dal comando. `KeyId` AWS KMS utilizza automaticamente la chiave che ha crittografato il testo cifrato. KMS
> Il `KeyId` parametro è obbligatorio quando si chiama `Decrypt` o si verifica con una chiave asimmetrica o si chiama con una KMS chiave. VerifyMacHMACKMS Queste richieste hanno esito negativo quando il valore del `KeyId` parametro è un alias che non punta più alla KMS chiave che ha eseguito l'operazione crittografica, ad esempio quando una chiave viene ruotata manualmente. Per evitare questo errore, è necessario tenere traccia e specificare la KMS chiave corretta per ogni operazione.

Per modificare la KMS chiave di destinazione di un alias, utilizzare l'UpdateAliasoperazione in. AWS KMS API Ad esempio, questo comando aggiorna l'`alias/TestKey`alias in modo che punti a una nuova KMS chiave. Poiché l'operazione non restituisce alcun output, l'esempio utilizza

l'ListAliasesoperazione per mostrare che l'alias è ora associato a una KMS chiave diversa e il `LastUpdatedDate` campo viene aggiornato. I ListAliases comandi utilizzano il `query`parametro in AWS CLI per ottenere solo l'`alias/TestKey`alias.

```
$ 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.

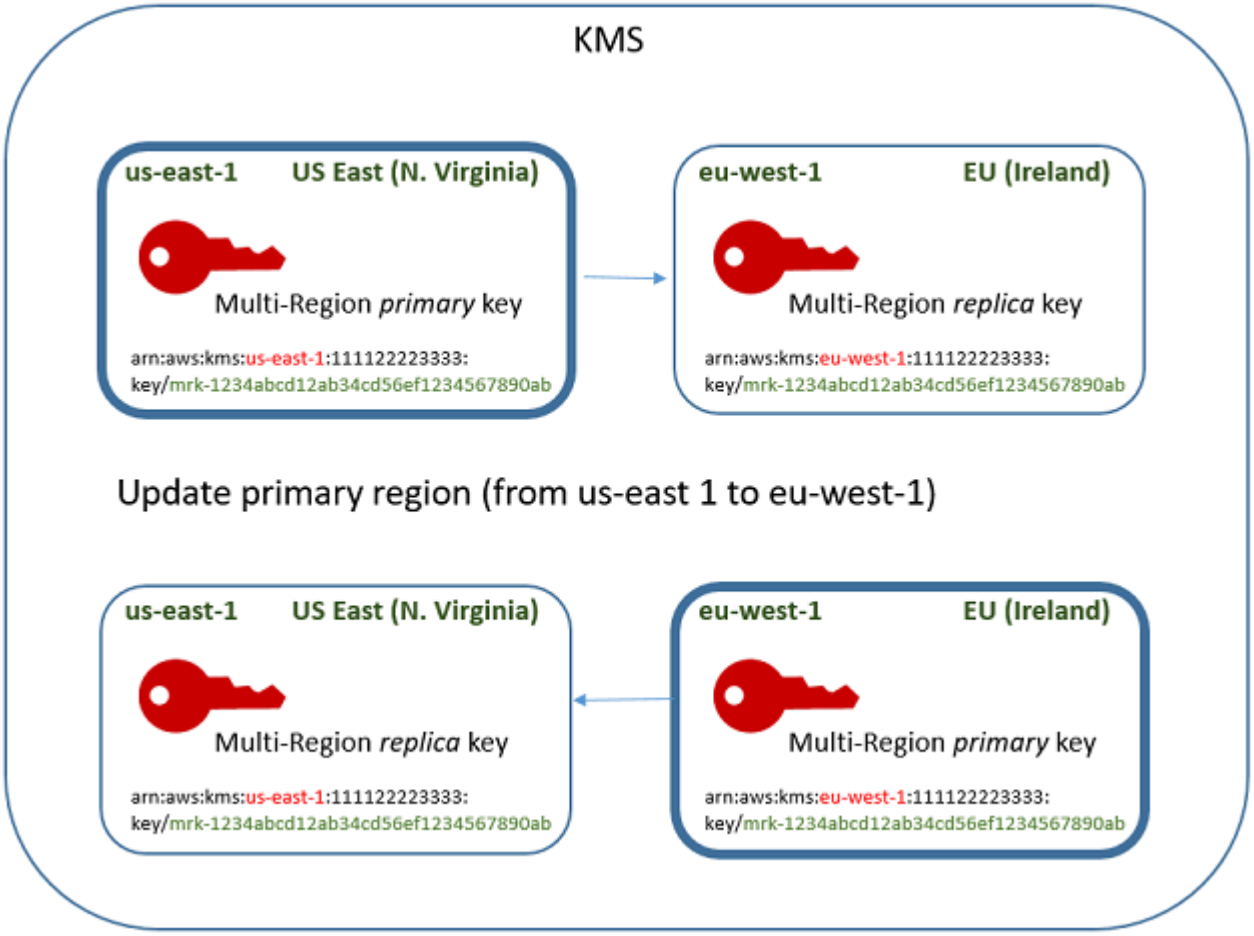
Lo stato **`Updating`** chiave

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 DescribeKeypotrebbero 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-est-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 effetto sulla chiave ARN di alcuna chiave multiregionale. 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'UpdatePrimaryRegionautorizzazione kms: in entrambe le regioni.

Puoi aggiornare la regione principale nella AWS KMS console o utilizzando l'UpdatePrimaryRegionoperazione.

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 AWS Management Console 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 il AWS KMS API

Per modificare la chiave primaria in un set di chiavi multiregionali correlate, utilizzare l'UpdatePrimaryRegionoperazione.

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 successo, questa operazione non restituisce alcun output, ma solo il codice di HTTP stato. Per vedere l'effetto, chiamate l'DescribeKeyoperazione 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 keys

L'eliminazione di un file AWS KMS key è distruttiva e potenzialmente pericolosa. Elimina il materiale chiave e tutti i metadati associati alla chiave ed è irreversibile. KMS Dopo l'eliminazione di una KMS chiave, non è più possibile decrittografare i dati crittografati con quella KMS chiave, il che significa che i dati diventano irrecuperabili. (Le uniche eccezioni sono le chiavi di replica multiregionali e asimmetriche e le chiavi con materiale chiave importato.) HMAC KMS Questo rischio è significativo per KMSle chiavi asimmetriche utilizzate per la crittografia in cui, senza avvisi o errori, 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

È consigliabile eliminare una KMS chiave solo quando si è certi di non averne più bisogno. Se non sei sicuro, valuta la possibilità di disabilitare la KMS chiave anziché eliminarla. È possibile riattivare una KMS chiave disabilitata e annullare l'eliminazione pianificata di una KMS chiave, ma non è possibile ripristinare una chiave eliminata. KMS

È possibile solamente pianificare l'eliminazione di una chiave gestita dal cliente. Non è possibile eliminare Chiavi gestite da AWS o Chiavi di proprietà di AWS.

Prima di eliminare una KMS chiave, 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, è necessario determinare l'utilizzo passato di una KMS chiave. Per assistenza, vai a Determinare l'utilizzo passato di una KMS chiave.

AWS KMS non elimina mai le KMS chiavi a meno che non ne pianifichi esplicitamente l'eliminazione e che il periodo di attesa obbligatorio scada.

Tuttavia, potresti scegliere di eliminare una KMS chiave per uno o più dei seguenti motivi:

- Per completare il ciclo di vita delle KMS chiavi che non ti servono più
- Per evitare il sovraccarico di gestione e i costi associati alla manutenzione delle chiavi inutilizzate KMS
- Per ridurre il numero di KMS chiavi che influiscono sulla quota di risorse KMSchiave

> **Note**
>
> Se chiudi il tuo Account AWS, KMS le tue chiavi diventano inaccessibili e non ti vengono più addebitate le spese.

AWS KMS registra una voce nel AWS CloudTrail registro quando pianifichi l'eliminazione della KMS chiave e quando la KMSchiave viene effettivamente eliminata.

Informazioni sul periodo di attesa

Poiché eliminare una KMS chiave 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 KMS chiave verrà eliminata, utilizzare l'DescribeKeyoperazione. Oppure nella AWS KMS console, nella pagina dei dettagli della KMS chiave, nella sezione Configurazione generale, vedi la Data di eliminazione pianificata. Assicurati di segnare il fuso orario.

Durante il periodo di attesa, lo KMS stato e lo stato della chiave sono In attesa di eliminazione.

- Una KMS chiave in attesa di eliminazione non può essere utilizzata in nessuna operazione crittografica.
- AWS KMS non ruota il materiale chiave delle KMS chiavi in attesa di eliminazione.

Al termine del periodo di attesa, AWS KMS elimina la KMS chiave, i relativi alias e tutti i metadati correlati. AWS KMS

La pianificazione dell'eliminazione di una KMS chiave potrebbe non influire immediatamente sulle chiavi dati crittografate dalla chiave. KMS Per informazioni dettagliate, consultare In che modo le chiavi inutilizzabili influiscono sulle chiavi dati KMS.

Usa il periodo di attesa per assicurarti di non aver bisogno della KMS chiave ora o in futuro. Puoi configurare un CloudWatch allarme Amazon per avvisarti se una persona o un'applicazione tenta di utilizzare la KMS chiave durante il periodo di attesa. Per recuperare la KMS chiave, puoi annullare l'eliminazione della chiave prima della fine del periodo di attesa. Al termine del periodo di attesa non è possibile annullare l'eliminazione della chiave e la KMS chiave AWS KMS viene eliminata.

Considerazioni speciali

Prima di pianificare l'eliminazione delle chiavi, consulta le seguenti considerazioni speciali sull'eliminazione delle chiavi per scopi speciali. KMS

Eliminazione delle chiavi asimmetriche KMS

Gli utenti autorizzati possono eliminare chiavi simmetriche o asimmetriche. KMS La procedura per pianificare l'eliminazione di queste KMS chiavi è la stessa per entrambi i tipi di chiavi. Tuttavia, poiché la chiave pubblica di una chiave asimmetrica KMS può essere scaricata e utilizzata all'esterno AWS KMS, l'operazione comporta rischi aggiuntivi significativi, in particolare per le KMS chiavi asimmetriche utilizzate per la crittografia (l'utilizzo della chiave è). `ENCRYPT_DECRYPT`

- Quando si pianifica l'eliminazione di una KMS chiave, lo stato della chiave cambia in In attesa di eliminazione e la KMS chiave non può essere utilizzata nelle operazioni KMS crittografiche. 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.
- Gli allarmi, i registri e altre strategie che rilevano il tentativo di utilizzo di una KMS chiave in attesa di eliminazione non possono rilevare l'uso della chiave pubblica all'esterno di. AWS KMS
- Quando la KMS chiave viene eliminata, tutte le AWS KMS azioni che coinvolgono quella chiave falliscono. KMS Tuttavia, gli utenti che dispongono della chiave pubblica possono continuare a utilizzarla per crittografare i messaggi. Questi testi cifrati non possono essere decrittati.

Se è necessario eliminare una chiave asimmetrica che utilizza una KMS chiave di`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. Quindi, valuta la possibilità di disabilitare la KMS chiave anziché eliminarla.

Il rischio rappresentato dall'eliminazione di una chiave asimmetrica è mitigato per le chiavi asimmetriche con materiale KMS chiave importato. KMS 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 nessun'altra KMS chiave. 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 di KMS chiavi con materiale chiave importato

L'eliminazione del materiale chiave di una KMS chiave con materiale chiave importato è temporanea e reversibile. Per ripristinare la chiave, reimporta il materiale della chiave.

Al contrario, l'eliminazione di una KMS chiave è irreversibile. Se si pianifica l'eliminazione di una chiave e il periodo di attesa richiesto scade, elimina AWS KMS in modo permanente e irreversibile la KMS chiave, il relativo materiale chiave e tutti i metadati associati alla chiave. KMS

Tuttavia, il rischio e le conseguenze dell'eliminazione di una KMS chiave con materiale chiave importato dipendono dal tipo («specifica chiave») della chiave. KMS

- Chiavi di crittografia simmetriche: se si elimina una chiave di crittografia simmetrica, tutti i testi cifrati rimanenti crittografati da quella KMS chiave sono irrecuperabili. Non è possibile creare una nuova chiave di crittografia simmetrica in grado di decrittografare i testi cifrati di una KMS chiave di crittografia simmetrica eliminata, anche se si dispone dello stesso materiale di chiave. KMS I metadati univoci di ogni chiave sono associati crittograficamente a ogni KMS testo cifrato simmetrico. Questa funzionalità di sicurezza garantisce che solo la KMS chiave che ha crittografato il testo cifrato simmetrico possa decrittografarlo, ma impedisce di ricreare una chiave equivalente. KMS
- Asimmetrico e HMAC chiavi: se disponi del materiale chiave originale, puoi creare una nuova chiave con le stesse proprietà crittografiche di una KMS chiave asimmetrica o eliminata. HMAC KMS AWS KMS genera RSA testi e firme cifrati standard, ECC firme e tag, che non includono funzionalità di sicurezza uniche. HMAC Inoltre, puoi utilizzare una HMAC chiave o la chiave privata di una coppia di chiavi asimmetrica all'esterno di. AWS

  Una nuova KMS chiave creata con lo stesso materiale asimmetrico o HMAC chiave avrà un identificatore di chiave diverso. Dovrai creare una nuova politica chiave, ricreare eventuali alias e aggiornare le IAM politiche e le sovvenzioni esistenti in modo che facciano riferimento alla nuova chiave.

Eliminazione delle chiavi da qualsiasi archivio di KMS chiavi AWS CloudHSM

Quando si pianifica l'eliminazione di una KMS chiave da un archivio AWS CloudHSM chiavi, lo stato della chiave cambia in In sospeso di eliminazione. La KMS chiave rimane nello stato In sospeso di eliminazione per tutto il periodo di attesa, anche se non è più disponibile perché è stato disconnesso l'archivio KMS chiavi personalizzato. Ciò consente di annullare l'eliminazione della KMS chiave in qualsiasi momento durante il periodo di attesa.

Allo scadere del periodo di attesa, AWS KMS elimina la KMS chiave 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 KMS chiave dal cluster AWS KMS e ne elimini il materiale chiave, i AWS CloudHSM cluster creati dai backup potrebbero contenere il materiale chiave eliminato. Per eliminare definitivamente il materiale chiave, utilizzate l'DescribeKeyoperazione per identificare la data di creazione della KMS chiave. Quindi elimina tutti i backup del cluster che potrebbero contenere quel materiale.

Quando si pianifica l'eliminazione di una KMS chiave da un archivio AWS CloudHSM chiavi, la KMS chiave diventa immediatamente inutilizzabile (fatta salva l'eventuale coerenza). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono influenzate fino a quando la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

Eliminazione delle KMS chiavi da un archivio di chiavi esterno

L'eliminazione di una KMS chiave da un archivio di chiavi esterno non ha alcun effetto sulla chiave esterna utilizzata come materiale chiave.

Quando si pianifica l'eliminazione di una KMS chiave da un archivio di chiavi esterno, lo stato della chiave cambia in In sospeso di eliminazione. La KMS chiave rimane nello stato In sospeso di eliminazione per tutto il periodo di attesa, anche se non è più disponibile perché è stato disconnesso l'archivio KMS chiavi esterno. Ciò consente di annullare l'eliminazione della KMS chiave in qualsiasi momento durante il periodo di attesa. Allo scadere del periodo di attesa, AWS KMS elimina la KMS chiave da. AWS KMS

Quando si pianifica l'eliminazione di una KMS chiave da un archivio chiavi esterno, la KMS chiave diventa immediatamente inutilizzabile (a seconda della coerenza finale). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono influenzate fino a quando la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

Controlla l'accesso all'eliminazione delle chiavi

Se si utilizzano IAM criteri per consentire le AWS KMS autorizzazioni, IAM le identità con accesso AWS amministratore (`"Action": "*"`) o accesso AWS KMS completo (`"Action": "kms:*"`) possono già pianificare e annullare l'eliminazione delle KMS chiavi. Per consentire agli amministratori delle chiavi di pianificare e annullare l'eliminazione delle chiavi nella politica delle chiavi, utilizza la AWS KMS console o il. AWS KMS API

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 IAM identità aggiungendo l'`kms:CancelKeyDeletion`autorizzazione `kms:ScheduleKeyDeletion` and alla politica chiave o a una politica. 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 AWS Management Console 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. Scegliete l'alias o l'ID della chiave di cui desiderate modificare le KMS 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).

Usando il AWS KMS API

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. Utilizzate il `aws kms put-key-policy`comando per applicare la politica della chiave alla chiave. KMS

Pianifica l'eliminazione della chiave

Le seguenti procedure descrivono come pianificare l'eliminazione delle chiavi e annullare l'eliminazione delle KMS chiavi di AWS KMS keys (chiavi) AWS KMS utilizzando AWS Management Console and the AWS KMS API.

> **Warning**
>
> L'eliminazione di una KMS chiave è distruttiva e potenzialmente pericolosa. Dovresti procedere solo quando sei sicuro di non aver più bisogno di usare la KMS chiave e di non averne bisogno in futuro. Se non sei sicuro, dovresti disabilitare la KMS chiave invece di eliminarla.

Prima di poter eliminare una KMS chiave, è necessario disporre dell'autorizzazione necessaria. 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 si pianifica l'eliminazione della KMS chiave e quando la KMSchiave viene effettivamente eliminata.

Utilizzo della console di AWS KMS

In AWS Management Console, è possibile pianificare e annullare l'eliminazione di più KMS chiavi contemporaneamente.

Per pianificare l'eliminazione di chiavi

1. Accedi a AWS Management Console 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. Seleziona la casella di controllo accanto alla KMS chiave 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 KMS chiavi 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 KMS chiave cambia in In attesa di eliminazione.

Usando il AWS KMS API

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 KMS chiave, è possibile 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'CancelKeyDeletionoperazione. Dopo aver annullato l'eliminazione in sospeso di una KMS chiave, lo stato della KMS chiave è`Disabled`. Per ulteriori informazioni sull'attivazione della KMS chiave, vedereAttivazione e disattivazione dei tasti.

Utilizzo della AWS KMS console

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. Seleziona la casella di controllo accanto alla KMS chiave che desideri recuperare.
5. Scegliere Key actions (Operazioni sulle chiavi), Cancel key deletion (Annulla eliminazione chiave).

Lo stato della KMS chiave cambia da In sospeso di eliminazione a Disabilitato. Per utilizzare la KMS chiave, è necessario abilitarla.

Usando il AWS KMS API

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 KMS chiave cambia da In sospeso di eliminazione a Disabilitato. Per utilizzare la KMS chiave, è necessario abilitarla.

Crea un allarme che rileva l'uso di una KMS chiave in attesa di eliminazione

Puoi combinare le funzionalità di AWS CloudTrail Amazon CloudWatch Logs e Amazon Simple Notification Service (AmazonSNS) per creare un CloudWatch allarme Amazon che ti avvisa quando qualcuno nel tuo account tenta di utilizzare una KMS chiave in attesa di eliminazione. Se ricevi questa notifica, potresti voler annullare l'eliminazione della KMS chiave e riconsiderare la tua decisione di eliminarla.

Le seguenti procedure creano un allarme che avvisa l'utente ogni volta che il messaggio di errore *`Key ARN`* `is pending deletion` "" viene scritto nei file di CloudTrail registro. Questo messaggio di errore indica che una persona o un'applicazione ha tentato di utilizzare la KMS chiave in un'operazione crittografica. Poiché la notifica è collegata al messaggio di errore, non viene attivata quando si utilizzano API operazioni consentite su KMS chiavi in attesa di eliminazione, ad `ListKeys` esempio, e. `CancelKeyDeletion` `PutKeyPolicy` Per visualizzare un elenco delle AWS KMS API operazioni che restituiscono questo messaggio di errore, vedere. Stati chiave delle AWS KMS chiavi

L'e-mail di notifica che ricevi non elenca la KMS chiave o l'operazione di crittografia. Puoi trovare queste informazioni nel tuo CloudTrail registro. 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 KMS chiave asimmetrica al di fuori di. AWS KMS Per informazioni sui rischi particolari legati all'eliminazione delle KMS chiavi asimmetriche utilizzate per la crittografia a chiave pubblica, inclusa la creazione di testi cifrati che non possono essere decifrati, 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	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
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 *`metric-name`* 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, potrebbe significare che è ancora necessaria una KMS chiave di cui è prevista l'eliminazione per crittografare o decrittografare i dati. In tal caso, annulla l'eliminazione della KMS chiave e riconsidera la tua decisione di eliminarla.

Determinare l'utilizzo passato di una KMS chiave

Prima di eliminare una KMS chiave, 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. Conoscere come veniva usata una KMS chiave in passato potrebbe aiutarvi a decidere se ne avrete bisogno o meno in futuro. Questo argomento suggerisce diverse strategie che possono aiutarti a determinare l'utilizzo passato di una KMS chiave.

> **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 KMS chiave asimmetrica al di fuori di. AWS KMS Per dettagli sui rischi particolari dell'eliminazione delle KMS chiavi asimmetriche utilizzate per la crittografia a chiave pubblica, inclusa la creazione di testi cifrati che non possono essere decifrati, vedere. Deleting asymmetric KMS keys

Argomenti

Esamina le KMS autorizzazioni delle chiavi per determinare l'ambito del potenziale utilizzo

Determinare chi o cosa ha attualmente accesso a una KMS chiave può aiutarti a determinare in che misura la KMS chiave è stata utilizzata e se è ancora necessaria. Per sapere come determinare chi o cosa ha attualmente accesso a una KMS chiave, vai aDeterminare l'accesso a AWS KMS keys.

Esamina AWS CloudTrail i log per determinare l'utilizzo effettivo

Potresti essere in grado di utilizzare una cronologia di utilizzo delle KMS chiavi per determinare se hai testi cifrati crittografati con una chiave particolare. KMS

Tutte le AWS KMS API attività vengono registrate nei AWS CloudTrail file di registro. Se hai creato un CloudTrail percorso nella regione in cui si trova la KMS chiave, puoi esaminare i file di CloudTrail

registro per visualizzare una cronologia di tutte le AWS KMS API attività relative a una determinata KMS chiave. 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, consultaRegistrazione delle AWS KMS API chiamate con AWS CloudTrail.

Gli esempi seguenti mostrano le voci di CloudTrail registro generate quando viene utilizzata una KMS chiave per proteggere un oggetto archiviato in Amazon Simple Storage Service (Amazon S3). In questo esempio, l'oggetto viene caricato su Amazon S3 utilizzando Protecting data using server-side encryption with KMS keys (-). SSE KMS Quando carichi un oggetto su Amazon S3 con SSE - KMS, specifichi la KMS chiave da utilizzare per proteggere l'oggetto. Amazon S3 utilizza AWS KMS GenerateDataKeyoperazione 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": "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": "2015-09-10T23:58:18Z",
  "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 specificata. KMS Quando esegui questa operazione, i tuoi file di CloudTrail registro 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 AWS KMS API attività vengono registrate da CloudTrail. Valutando queste voci di registro, potresti essere in grado di determinare l'utilizzo passato di una particolare KMS chiave e questo potrebbe aiutarti a determinare se desideri eliminarla o meno.

Per vedere altri esempi di come AWS KMS API l'attività viene visualizzata nei file di CloudTrail registro, vai aRegistrazione delle AWS KMS API chiamate 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 KMS chiave 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 KMS chiave cambia in attesa di importazione e la KMS chiave non può essere utilizzata in alcuna operazione crittografica finché non si reimporta lo stesso materiale chiave. (Non è possibile importare altro materiale chiave nella chiave.) KMS

Oltre a disabilitare la KMS chiave e revocare le autorizzazioni, l'eliminazione del materiale chiave può essere utilizzata come strategia per interrompere rapidamente, ma temporaneamente, l'uso della chiave. KMS Al contrario, la pianificazione dell'eliminazione di una KMS chiave con materiale chiave importato interrompe rapidamente anche l'utilizzo della chiave. KMS Tuttavia, se l'eliminazione non viene annullata durante il periodo di attesa, la chiave, il materiale KMS chiave e tutti i metadati chiave vengono eliminati definitivamente. Per informazioni dettagliate, consultare Deleting KMS keys with imported key material.

Per eliminare il materiale chiave, è possibile utilizzare la AWS KMS console o l'operazione. DeleteImportedKeyMaterialAPI 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 il materiale chiave, la KMS chiave senza materiale chiave diventa immediatamente inutilizzabile (a seconda della coerenza finale). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono influenzate fino a quando la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

Utilizzo della console AWS KMS

È possibile utilizzare la AWS KMS console per eliminare il materiale chiave.

1. Accedi AWS Management Console 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:

   - Selezionate la casella di controllo relativa a una KMS chiave con materiale chiave importato. Scegliere Key actions (Operazioni della chiave), Delete key material (Elimina materiale della chiave).
   - Scegliete l'alias o l'ID chiave di una KMS chiave con materiale chiave importato. Scegli la tab Materiale chiave e quindi scegli Elimina materiale chiave.

5. Confermare che si intende eliminare il materiale della chiave, quindi selezionare Delete key material (Elimina materiale chiave). Lo stato della KMS chiave, che corrisponde allo stato della chiave, cambia in In attesa di importazione.

Usando il AWS KMS API

Per utilizzare l'opzione AWS KMS APIper eliminare il materiale chiave, invia una DeleteImportedKeyMaterialrichiesta. L'esempio seguente mostra come eseguire questa operazione con l'AWS CLI.

Sostituisci *1234abcd-12ab-34cd-56ef-1234567890ab* con l'ID della KMS chiave di cui desideri eliminare il materiale chiave. È possibile utilizzare l'ID della KMS chiave oppure, ARN ma non è possibile utilizzare un alias per questa operazione.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

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 KMS chiavi simmetriche, che non possono essere scaricate, le chiavi dati ti vengono restituite per essere utilizzate all'esterno di. AWS KMS

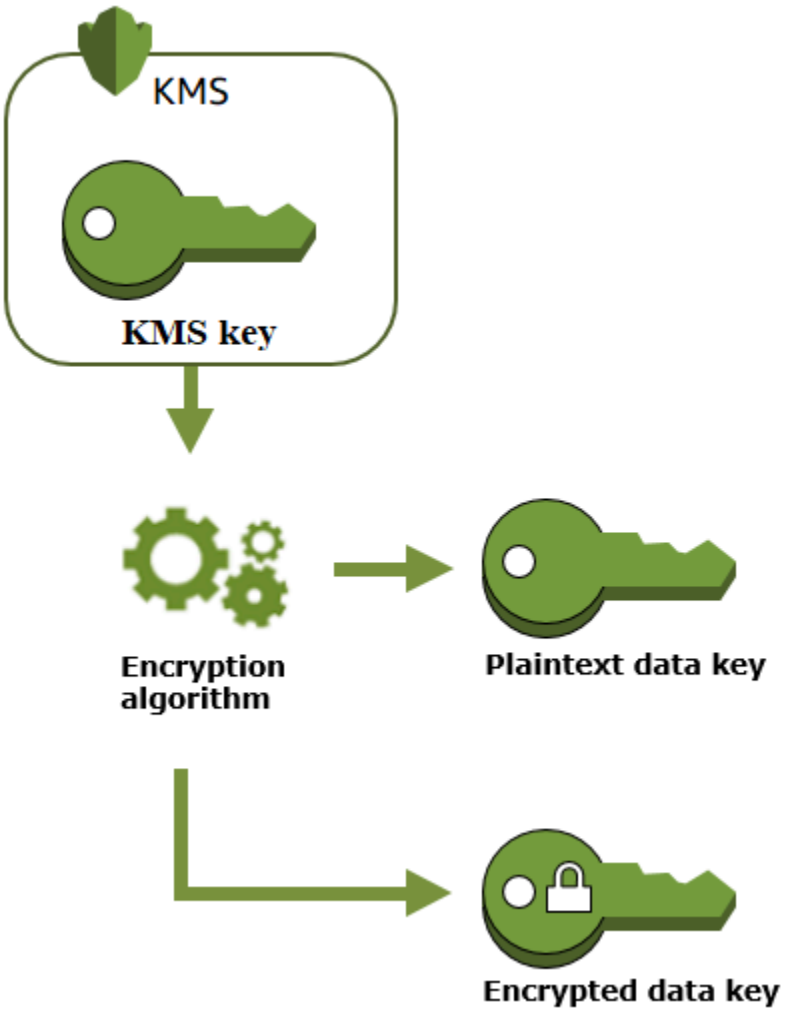
Quando AWS KMS genera chiavi dati, restituisce una chiave dati in testo semplice per l'uso immediato (opzionale) e una copia crittografata della chiave dati che puoi archiviare in sicurezza con i dati. Quando sei pronto per decrittografare i dati, chiedi innanzitutto di AWS KMS decrittografare la chiave 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

Per creare una chiave dati, richiama l'GenerateDataKeyoperazione. AWS KMS genera la chiave dati. Quindi crittografa una copia della chiave dati con una chiave di crittografia KMS simmetrica specificata dall'utente. L'operazione restituisce una copia in testo semplice della chiave dati e la copia della chiave dati crittografata sotto la chiave. KMS L'immagine seguente mostra questa operazione.



AWS KMS supporta anche l'GenerateDataKeyWithoutPlaintextoperazione, che restituisce solo una chiave dati crittografata. Quando devi usare la chiave dati, chiedi di AWS KMS decrittografarla.

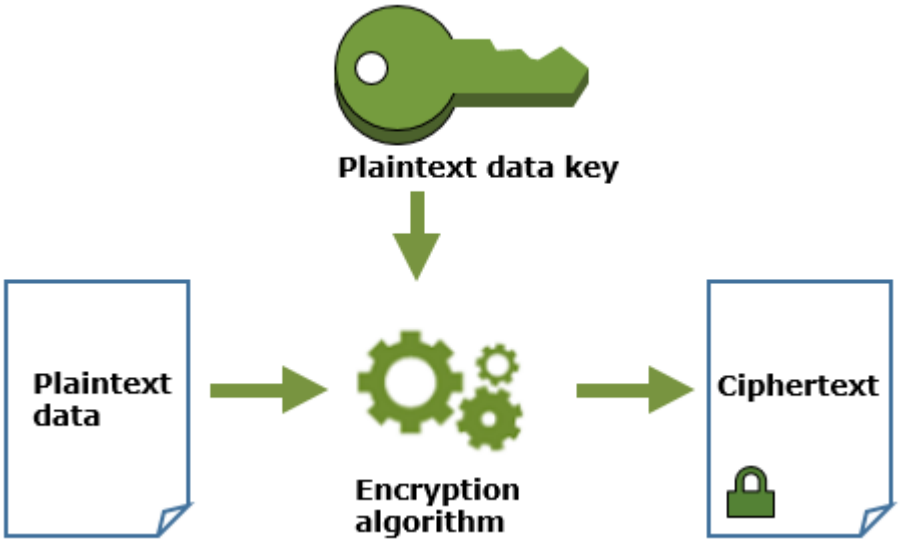
Come funzionano le operazioni crittografiche con chiavi dati

I seguenti argomenti spiegano come funzionano le chiavi dati generate da un'GenerateDataKeyWithoutPlaintextoperazione GenerateDataKeyor.

Crittografia dei dati con una chiave di dati

AWS KMS non può utilizzare una chiave dati per crittografare i dati. È tuttavia possibile utilizzare la chiave dati all'esterno di AWS KMS, ad esempio utilizzando Open SSL o una libreria crittografica come. AWS Encryption SDK

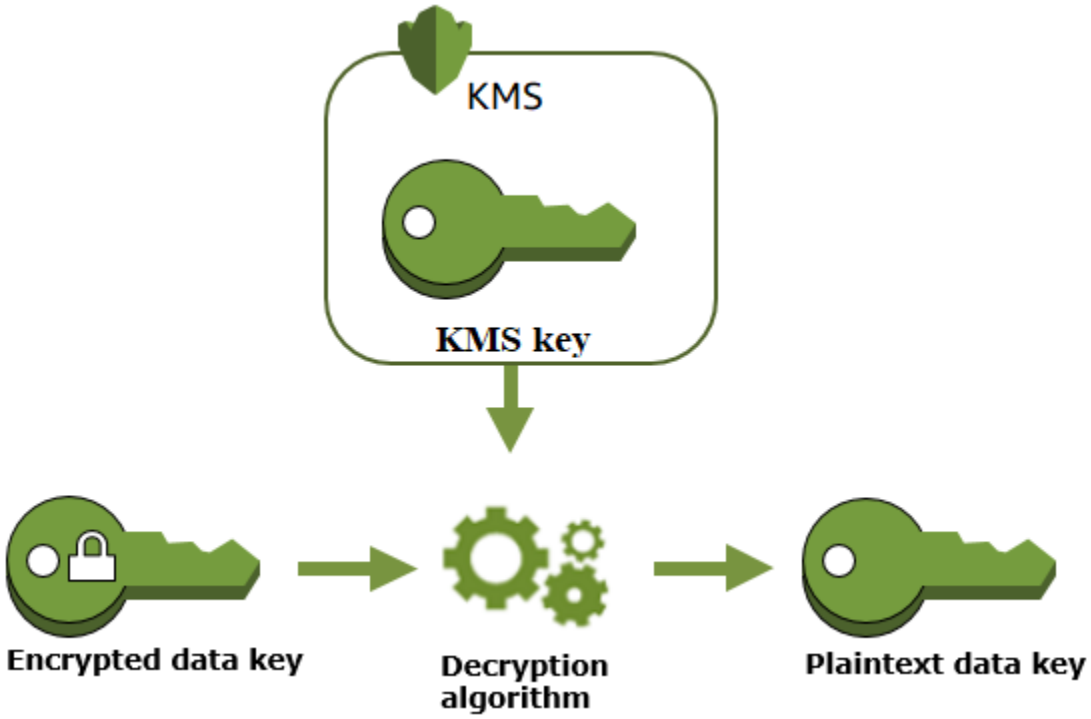
Dopo aver utilizzato la chiave di dati in testo normale per crittografare i dati, eliminarla dalla memoria il prima possibile. È possibile archiviare la chiave di dati crittografati con i dati crittografati in totale sicurezza, in modo che sia disponibile per decrittografare i dati.



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 tua KMS chiave per decrittografare la chiave dati e quindi restituisce la chiave di 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 inutilizzabili influiscono sulle chiavi dati KMS

Quando una KMS chiave diventa inutilizzabile, l'effetto è quasi immediato (soggetto alla coerenza finale). Lo stato della KMS chiave cambia in base alla nuova condizione e tutte le richieste di utilizzo della KMS chiave nelle operazioni crittografiche hanno esito negativo.

Tuttavia, l'effetto sulle chiavi dati crittografate dalla KMS chiave e sui dati crittografati dalla chiave dati viene ritardato fino a quando la KMS chiave non viene riutilizzata, ad esempio per decrittografare la chiave dati.

KMSle chiavi possono diventare inutilizzabili per diversi motivi, tra cui le seguenti azioni che è possibile eseguire.

- Disattivazione della chiave KMS
- Pianificazione della cancellazione della KMS chiave
- Eliminare il materiale chiave da una KMS chiave con materiale chiave importato o lasciare scadere il materiale chiave importato.
- Disconnettere l'archivio delle AWS CloudHSM chiavi che ospita la KMS chiave o eliminare la chiave dal AWS CloudHSM cluster che funge da materiale chiave per la chiave. KMS

- Disconnessione dell'archivio di chiavi esterno che ospita la KMS chiave o qualsiasi altra azione che interferisca con le richieste di crittografia e decrittografia al proxy dell'archivio chiavi esterno, inclusa l'eliminazione della chiave esterna dal relativo gestore di chiavi esterno.

Questo effetto è particolarmente importante per molti utenti Servizi AWS che utilizzano chiavi di dati per proteggere le risorse gestite dal servizio. L'esempio seguente utilizza Amazon Elastic Block Store (AmazonEBS) e Amazon Elastic Compute Cloud (AmazonEC2). Diversi Servizi AWS utilizzano le chiavi dati 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. Si crea un EBS volume crittografato e si specifica una KMS chiave per proteggerlo. Amazon EBS chiede AWS KMS di utilizzare la tua KMS chiave 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 EBS volume a un'EC2istanza, Amazon EC2 usa la tua KMS chiave per decrittografare la chiave dati crittografata del EBS volume. Amazon EC2 utilizza la chiave dati nell'hardware Nitro, che è responsabile della crittografia di tutti gli I/O del disco sul volume. EBS La chiave dati persiste nell'hardware Nitro mentre il EBS volume è collegato all'istanza. EC2
3. Si esegue un'azione che rende la KMS chiave inutilizzabile. Ciò non ha alcun effetto immediato sull'EC2istanza o sul EBS volume. Amazon EC2 utilizza la chiave dati, non la KMS chiave, per crittografare tutti gli I/O del disco mentre il volume è collegato all'istanza.
4. Tuttavia, quando il EBS volume crittografato viene scollegato dall'EC2istanza, Amazon EBS rimuove la chiave dati dall'hardware Nitro. La prossima volta che il EBS volume crittografato viene collegato a un'EC2istanza, l'allegato non riesce perché Amazon EBS non può utilizzare la KMS chiave per decrittografare la chiave dati crittografata del volume. Per utilizzare nuovamente il EBS volume, devi rendere nuovamente utilizzabile la KMS chiave.

Genera coppie di chiavi di dati

Una chiave asimmetrica rappresenta una coppia di KMS 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 AWS KMS tra due peer.

A differenza delle coppie di chiavi di dati SSL generate da strumenti come Open, AWS KMS protegge la chiave privata in ogni coppia di chiavi di dati con una chiave di crittografia KMS simmetrica specificata AWS KMS dall'utente. 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

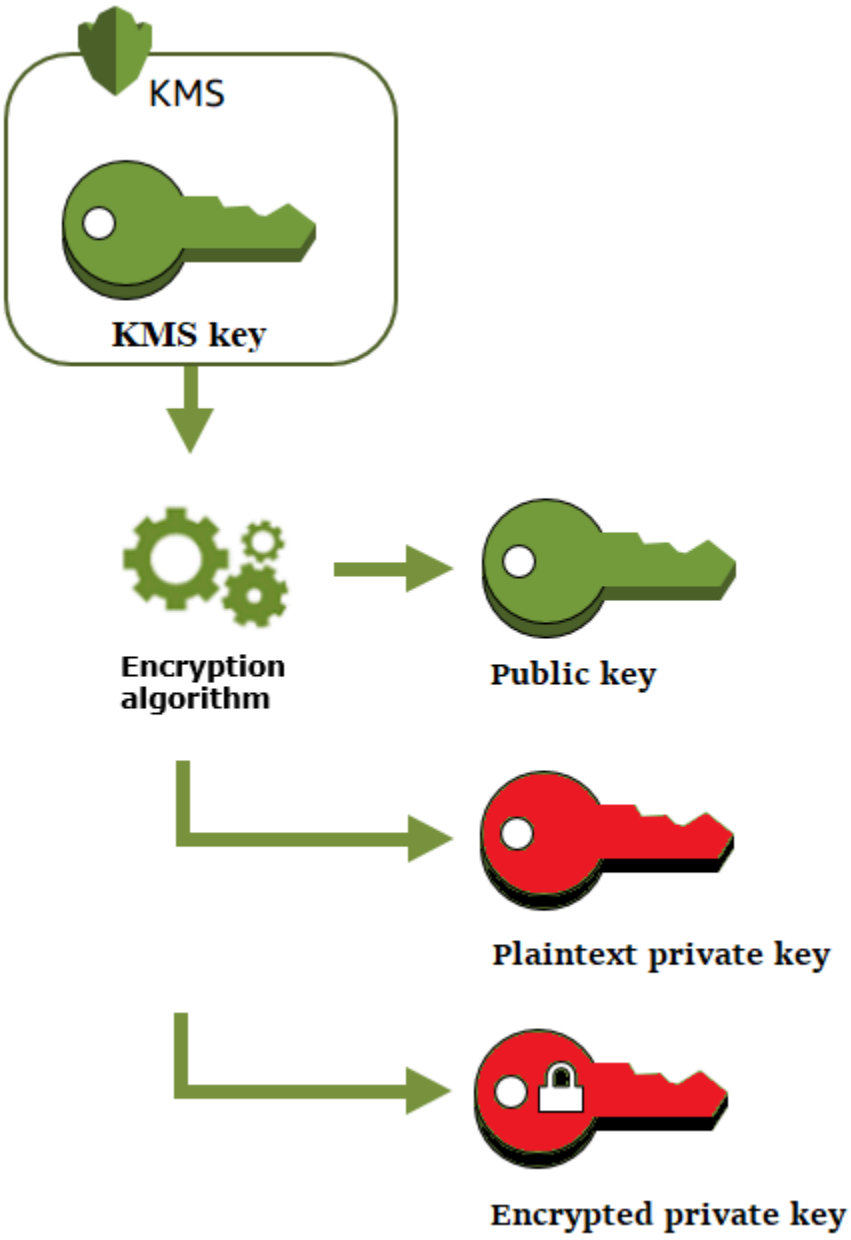
Creare una coppia di chiave di dati

Per creare una coppia di chiavi di dati, chiama le GenerateDataKeyPairWithoutPlaintextoperazioni GenerateDataKeyPairo. Specificate la KMSchiave di crittografia simmetrica che desiderate 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.



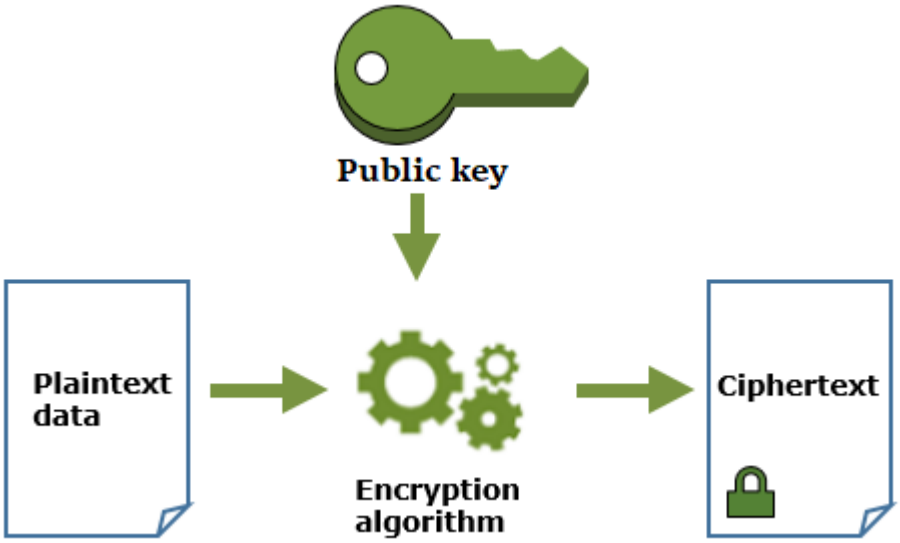
Come funzionano le operazioni crittografiche con coppie di chiavi di dati

I seguenti argomenti spiegano quali operazioni crittografiche è possibile eseguire con le coppie di chiavi di dati generate da un'GenerateDataKeyPairWithoutPlaintextoperazione GenerateDataKeyPairor 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.

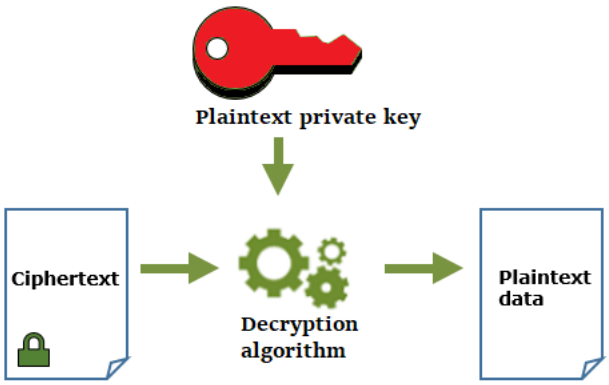


Decrittografia dei dati con una coppia di chiave di dati

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.



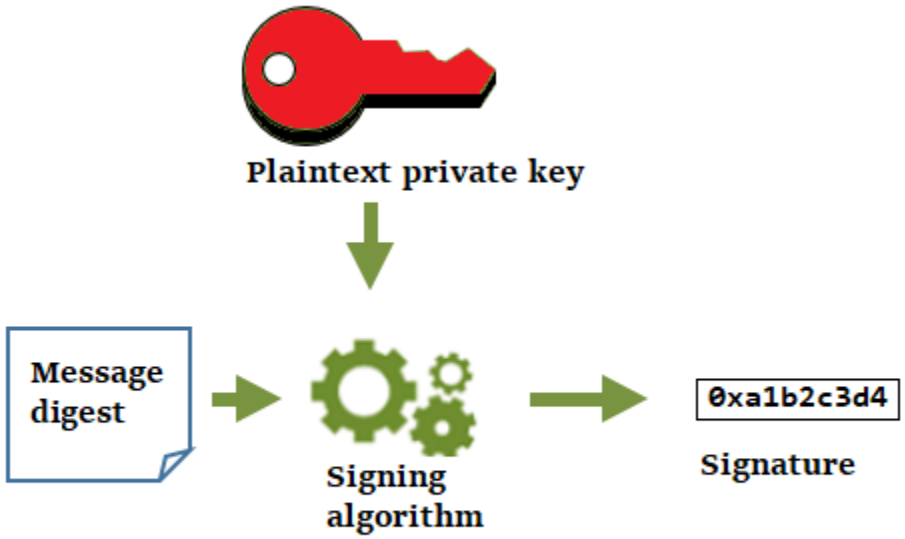
Firmare messaggi con una coppia di chiavi di dati

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 tua KMS chiave 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, crea un message digest utilizzando una funzione hash crittografica, ad esempio il comando in Open. `dgst`SSL 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.

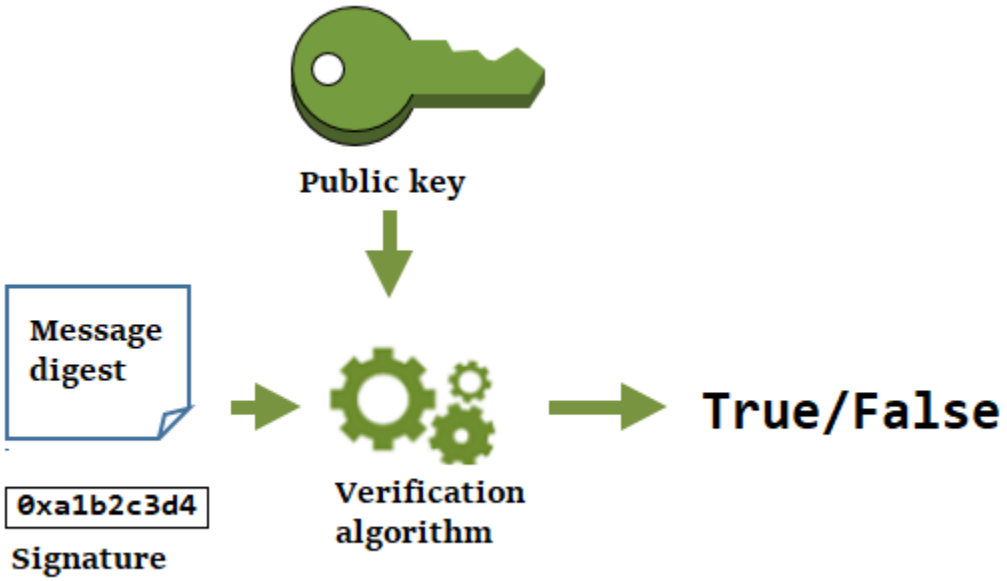


Verificare una firma con una coppia di chiavi di dati

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 interlocutore 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 asimmetrica, la KMS chiave privata viene creata e non viene mai crittografata. AWS KMS AWS KMS Per utilizzare la chiave privata, è necessario chiamare. AWS KMSÈ possibile utilizzare la chiave pubblica interna AWS KMS chiamando le AWS KMS API operazioni. 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 asimmetrica internaKMS, beneficiate dell'autenticazione AWS KMS, 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**
>
> Cerchi chiavi o SSH chiavi dati? 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 KMS chiave di crittografia simmetrica, vedi. GenerateDataKeyPair Per assistenza su come scaricare la chiave pubblica associata a un'EC2istanza 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

Per proteggere KMS le tue chiavi, AWS KMS fornisce controlli di accesso, crittografia autenticata e registri dettagliati di ogni operazione. AWS KMS consente inoltre di impedire l'uso delle KMS chiavi, 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 politiche chiave e IAMle politiche che controllano l'accesso alla KMS chiave 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 AWS KMS anche se non dispone dell'autorizzazione per crittografare i dati o verificare le firme con la KMS chiave.

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 KMS chiave con un `KeyUsage` of`SIGN_VERIFY`, l' AWS KMS operazione ha esito negativo. Tuttavia, se si crittografano i dati all'esterno AWS KMS con una chiave pubblica proveniente da una KMS chiave con un `KeyUsage` of `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 AWS KMS al di fuori di. AWS KMS Se si crittografano i dati con la chiave pubblica proveniente da una KMS chiave esterna e si utilizza un algoritmo di AWS KMS crittografia che AWS KMS non supporta, i dati non possono essere decrittografati.

Disabilitazione ed eliminazione delle chiavi KMS

Le azioni che è possibile intraprendere per impedire l'uso della KMS chiave in un'operazione crittografica interna non impediscono a nessuno di utilizzare la chiave pubblica all'esterno. AWS KMS AWS KMS Ad esempio, la disabilitazione di una KMS chiave, la pianificazione dell'eliminazione di una KMS chiave, l'eliminazione di una KMS chiave o l'eliminazione del materiale chiave da una chiave non hanno alcun effetto su una KMS chiave pubblica esterna a. AWS KMS Se si elimina una KMS chiave asimmetrica o si elimina o si perde 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 KMS chiavi asimmetrica nella AWS KMS console o utilizzando l'operazione. GetPublicKey Per scaricare la chiave pubblica, è necessario disporre dell'`kms:GetPublicKey`autorizzazione sulla chiave asimmetrica. KMS

La chiave pubblica AWS KMS restituita è una chiave pubblica X.509 DER con codifica, nota anche come `SubjectPublicKeyInfo` ()SPKI, come definita in 5280. RFC Quando si utilizza HTTP API o il, il valore è codificato in AWS CLI Base64. Altrimenti, non è codificato in Base64.

Per scaricare la chiave pubblica da una coppia di KMS key pair asimmetrica, sono necessarie le autorizzazioni. `kms:GetPublicKey` Per ulteriori informazioni sulle AWS KMS autorizzazioni, vedere. Riferimento per le autorizzazioni

Utilizzo della console AWS KMS

Puoi utilizzare il AWS Management Console per visualizzare, copiare e scaricare la chiave pubblica da una chiave asimmetrica KMS del tuo. Account AWS Per scaricare la chiave pubblica da una chiave asimmetrica KMS in modo diverso, usa il. Account AWS AWS KMS API

1. Accedi a AWS Management Console 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. Scegliete l'alias o l'ID della chiave asimmetrica. KMS
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).

Usando il AWS KMS API

L'GetPublicKeyoperazione restituisce la chiave pubblica in una chiave asimmetricaKMS. 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 KMS chiave, utilizza l'ID della chiave, la chiave ARN, il nome alias o l'alias. ARN Quando utilizzi un nome alias, aggiungi il prefisso alias/. Per specificare una KMS chiave in un'altra Account AWS, è necessario utilizzarne la chiave ARN o l'alias. ARN

Prima di eseguire questo comando, sostituite il nome alias di esempio con un identificatore valido per la chiave. KMS Per eseguire questo comando, è necessario disporre `kms:GetPublicKey` delle autorizzazioni sulla 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 KMS chiavi asimmetriche, 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 SM2 chiavi (solo regioni della Cina)

Ricevere segreti condivisi offline

È possibile scaricare la chiave pubblica della propria coppia di ECC chiavi per utilizzarla in operazioni offline, ovvero operazioni esterne a AWS KMS.

La seguente SSL procedura dettagliata di Open illustra un metodo per derivare un segreto condiviso al di fuori dell' AWS KMS utilizzo della chiave pubblica di una coppia di chiavi e di una ECC KMS chiave privata creata con Open. SSL

1. Crea una ECC key pair in Open SSL 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 ECC chiavi per l'accordo chiave 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. Ricava un segreto condiviso in Open SSL utilizzando la chiave privata in Open SSL e la KMS chiave 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 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`, al SignAPI, AWS KMS utilizza l'ID distintivo predefinito1234567812345678, definito OSCCA in GM/T 0009-2012. 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 SM2DSA message digest avvengono automaticamente. Non tutti i provider di crittografia implementano SM2 allo stesso modo. Alcune librerie, come SSL le versioni Open 1.1.1 e successive, eseguono queste azioni automaticamente. AWS KMS ha confermato questo comportamento durante i test con la SSL versione 3.0 di Open. 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 passare AWS KMS alla firma, utilizza il `calculateSM2Digest` metodo. Il `calculateSM2Digest` metodo genera un digest dei messaggi con l'algoritmo di SM3 hashing. GetPublicKeyAPIRestituisce 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. GetPublicKeyAPIRestituisce 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 nel formato .1. SM2PKE ASN GetPublicKeyAPIRestituisce la chiave pubblica in formato binario. È necessario analizzare la chiave binaria in un file Java PublicKey. Fornisci alla chiave pubblica analizzata il testo normale che desideri crittografare.

  Se non siete sicuri di dover eseguire la conversione, utilizzate la seguente SSL operazione Open per testare il formato del testo cifrato. Se l'operazione fallisce, è necessario convertire il testo cifrato nel formato .1. ASN

  ```
  openssl asn1parse -inform DER -in ciphertext.der
  ```

Per impostazione predefinita, la `SM2OfflineOperationHelper` classe utilizza l'ID distintivo predefinito quando genera i digest `1234567812345678` dei messaggi per le operazioni. SM2DSA

```
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 a monitorare le KMS chiavi, tuttavia, create 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 consiste nel monitorare le KMS chiavi nel tempo per stabilire una linea di base per il normale AWS KMS utilizzo e le aspettative nell'ambiente. Durante il monitoraggio delle KMS chiavi, archivia i dati di monitoraggio cronologici in modo da poterli confrontare con i dati attuali, identificare modelli e anomalie normali e ideare metodi per risolvere i problemi.

Ad esempio, puoi monitorare le AWS KMS API attività e gli eventi 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 APIattività per le operazioni sul piano dati. Si tratta di operazioni crittografiche che utilizzano una KMS chiave, ad esempio Decrypt, Encrypt e. ReEncryptGenerateDataKey
- AWS KMS APIattività per le operazioni del piano di controllo che sono importanti per l'utente. Queste operazioni gestiscono una KMS chiave e potresti voler monitorare quelle che modificano la disponibilità di una KMS chiave (come ScheduleKeyDeletion,, CancelKeyDeletion, DisableKeyEnableKeyImportKeyMaterial, e DeleteImportedKeyMaterial) o modificano il controllo di accesso di una KMS chiave (come PutKeyPolicyand RevokeGrant).
- Altre AWS KMS metriche (come la quantità di tempo rimanente alla scadenza del materiale chiave importato) ed eventi (come la scadenza del materiale chiave importato o l'eliminazione o la rotazione della KMS chiave).

Strumenti di monitoraggio

AWS fornisce vari strumenti che è possibile utilizzare per monitorare le KMS chiavi. 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

Puoi utilizzare i seguenti strumenti di monitoraggio automatizzato per controllare KMS le tue chiavi e segnalare quando qualcosa è cambiato.

- AWS CloudTrail Monitoraggio dei registri: 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 (AmazonSNS) o a una politica di Amazon EC2 Auto 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 KMS le chiavi 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 KMS le chiavi 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 delle KMS chiavi di monitoraggio 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 della AWS KMS console Chiavi gestite da AWSe delle chiavi gestite dal cliente per visualizzare le seguenti informazioni su ciascuna KMS chiave:

- ID chiave
- Stato
- Data di creazione
- Data di scadenza (per KMS le chiavi con materiale chiave importato)
- Origin
- ID dell'archivio chiavi personalizzato (per KMS le chiavi negli archivi di chiavi personalizzati)

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 AWS KMS API chiamate 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 API chiamate AWS KMS come eventi, incluse le chiamate dalla AWS KMS console AWS KMS APIs, dai AWS CloudFormation modelli, da AWS Command Line Interface (AWS CLI) e AWS Tools for PowerShell.

CloudTrail registra tutte le AWS KMS operazioni, incluse le operazioni di sola lettura, come ListAliasesand GetKeyRotationStatus, le operazioni che gestiscono le KMS chiavi, come and, CreateKeye le operazioni crittografiche PutKeyPolicy, come e 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 su più account sulle KMS chiavi vengono registrate sia nell'account 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 GetKeyPolicycrittografia. Per semplificare la ricerca delle voci di CloudTrail registro per KMS chiavi particolari, AWS KMS aggiunge la chiave ARN della KMS chiave interessata al `responseElements` campo nelle voci di registro per alcune operazioni di gestione delle AWS KMS chiavi, anche quando l'APIoperazione non restituisce la chiave. ARN

Sebbene per impostazione predefinita, tutte AWS KMS le azioni vengano registrate come CloudTrail eventi, è possibile escludere AWS KMS le azioni da una CloudTrail traccia. Per informazioni dettagliate, consultare Esclusione di AWS KMS eventi da un percorso.

Ulteriori informazioni:

- Per esempi di CloudTrail log di AWS KMS operazioni per un'enclave AWS Nitro, vedi. Richieste di monitoraggio per enclavi Nitro

Argomenti

Ricerca AWS KMS delle voci di registro in CloudTrail

Per cercare le voci di CloudTrail registro, usa la CloudTrail console o l'CloudTrail LookupEventsoperazione. 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 KMS chiave. Questi valori degli attributi potrebbero essere nulli nelle CloudTrail voci precedenti per le seguenti operazioni: CreateAlias, CreateGrant, DeleteAlias, DeleteImportedKeyMaterial, ImportKeyMaterial, ReplicateKey, RetireGrantRevokeGrantUpdateAlias, e. 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 KMS chiave particolare, ad esempio `CreateKey` and`EnableKey`, ma no`ListKeys`.
Nome risorsa (`ResourceName`)	Chiave ARN (o ID chiave e chiaveARN)	Operazioni di gestione che modificano una KMS chiave particolare, ad esempio `CreateKey` and`EnableKey`, ma non`ListKeys`.

Per facilitare la ricerca delle voci di registro per le operazioni di gestione su KMS chiavi particolari, AWS KMS registra la chiave ARN della KMS chiave interessata nell'`responseElements.keyId`elemento della voce di registro, anche quando l' AWS KMS APIoperazione non restituisce la chiaveARN.

Ad esempio, una chiamata riuscita all'DisableKeyoperazione non restituisce alcun valore nella risposta, ma anziché un valore nullo, il `responseElements.keyId` valore nella voce di DisableKey registro include la chiave ARN della KMS chiave disabilitata.

Questa funzionalità è stata aggiunta a dicembre 2022 e influisce sulle seguenti voci di CloudTrail registro: CreateAliasCreateGrantDeleteAlias, DeleteKey, DisableKey, EnableKey, EnableKeyRotation, ImportKeyMaterial, RotateKey, SynchronizeMultiRegionKey, TagResource, UntagResourceUpdateAlias, e 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 della politica delle chiavi e l'uso delle KMS chiavi da parte dei AWS servizi che agiscono 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 tue 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 AWS Management Console nella Guida per l' AWS CloudTrail utente.
- In CloudTrail API, utilizzare l'PutEventSelectorsoperazione. 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 oppureAPI, l' CloudTrail`PutEventSelectors`APIoperazione risultante viene registrata anche nei registri. CloudTrail Se AWS KMS gli eventi non compaiono nei tuoi CloudTrail log, cerca un `PutEventSelectors` evento con l'`ExcludeManagementEventSources`attributo 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 KMS chiave di cui è stata pianificata 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 AWS Nitro Enclaves, vedi. Richieste di monitoraggio per enclavi Nitro

Argomenti

- CancelKeyDeletion
- ConnectCustomKeyStore
- CreateAlias
- CreateCustomKeyStore
- CreateGrant
- CreateKey
- Decrypt
- DeleteAlias
- DeleteCustomKeyStore
- DeleteExpiredKeyMaterial

- DeleteImportedKeyMaterial
- DeleteKey
- DescribeCustomKeyStores
- DescribeKey
- DisableKey
- DisableKeyRotation
- DisconnectCustomKeyStore
- EnableKey
- EnableKeyRotation
- Crittografa
- 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
- Esempio uno di Amazon EC2
- Esempio due di Amazon EC2

CancelKeyDeletion

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione CancelKeyDeletion. Per informazioni sull'eliminazione delle AWS KMS keys, consulta Eliminare un AWS KMS keys.


```
{
    "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 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": "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'CreateAliasoperazione. L'elemento `resources` include i campi per l'alias e le risorse della chiave KMS. Per informazioni sulla creazione di alias in AWS KMS, consulta 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 log AWS CloudTrail generata chiamando l'operazione `CreateCustomKeyStore` in un archivio delle chiavi di AWS CloudHSM. 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'CreateGrantoperazione. Per informazioni sulla creazione delle concessioni in AWS KMS, consulta 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 AWS CloudTrail voci di registro relative all'CreateKeyoperazione.

Una voce di `CreateKey` registro può derivare da una `CreateKey` richiesta o dall'`CreateKey`operazione relativa a una ReplicateKeyrichiesta.

L'esempio seguente mostra una voce di CloudTrail registro per un'CreateKeyoperazione che crea una chiave di crittografia KMS simmetrica. Per informazioni sulla creazione di KMS chiavi, vedere. 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'`CreateKey`operazione che crea una KMS chiave di crittografia simmetrica in un AWS CloudHSM archivio 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": "2021-10-14T17:39:50Z",
    "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'`CreateKey`operazione che crea una KMS chiave 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,
        "customKeyStoreId": "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-8ff4-7736bbeb1316",
    "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"
}
```


Decrypt

Questi esempi mostrano AWS CloudTrail voci di registro per l'operazione Decrypt.

La voce di CloudTrail registro di un'`Decrypt`operazione include sempre le `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 in un KMS AWS CloudHSM archivio di chiavi
- Decifrare con una KMS chiave in un archivio di chiavi esterno
- Decifra l'errore con una KMS chiave in un archivio di chiavi esterno

Decrittografia con una chiave crittografica simmetrica standard

Di seguito è riportato un esempio di voce di CloudTrail registro per un'`Decrypt`operazione con una chiave di crittografia simmetrica standard.


```
{
    "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",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
```

```
    },
    "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"
}
```

Errore di decrittografia con una chiave crittografica simmetrica standard

L'esempio seguente di voce di CloudTrail registro registra un'`Decrypt`operazione non riuscita con una chiave di crittografia simmetrica standard. KMS L'eccezione (`errorCode`) e il messaggio di errore (`errorMessage`) sono inclusi per aiutarti a risolvere l'errore.

In questo caso, la chiave di crittografia simmetrica specificata nella `Decrypt` richiesta non era la KMS chiave di crittografia KMS simmetrica 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 in un KMS AWS CloudHSM archivio di chiavi

L'esempio seguente di voce di CloudTrail registro registra un'`Decrypt`operazione con una KMS chiave in un AWS CloudHSM archivio chiavi. Tutte le voci di registro per le operazioni crittografiche con una KMS chiave in un archivio di chiavi personalizzato includono un `additionalEventData` campo con `customKeyStoreId` e`backingKeyId`. Il valore restituito nel `backingKeyId` campo è l'`id`attributo HSM chiave Cloud. `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"
}
```


Decifrare con una KMS chiave in un archivio di chiavi esterno

L'esempio seguente di voce di CloudTrail registro registra un'`Decrypt`operazione con una KMS chiave in un archivio di chiavi esterno. Oltre a `customKeyStoreId`, 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"
}
```

Decifra l'errore con una KMS chiave in un archivio di chiavi esterno

L'esempio seguente di voce di CloudTrail registro registra una richiesta non riuscita di un'`Decrypt`operazione con una KMS chiave 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"
}
```

DeleteAlias

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'DeleteAliasoperazione. 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione `DeleteCustomKeyStore`. 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": {
        "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"
}
```


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 KMSregistra 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 log AWS CloudTrail generata quando AWS KMS elimina il materiale della chiave scaduto.


```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-01T16:00:00Z",
    "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"
    }
}
```

DeleteImportedKeyMaterial

Se importi materiale chiave in una chiave KMS, puoi eliminare il materiale chiave importato in qualsiasi momento utilizzando l'DeleteImportedKeyMaterialoperazione. 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, vedi Eliminare il materiale chiave importato.

L'esempio seguente illustra una voce di log AWS CloudTrail generata per l'operazione `DeleteImportedKeyMaterial`.

```
{
```

```
    "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": "2022-10-04T21:43:33Z",
    "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"
    },
    "responseElements": {
        "keyId":"&example-key-arn-1;"
    },
    "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"
}
```


DeleteKey

Questi esempi mostrano AWS CloudTrail voce di registro generata quando viene eliminata una KMS chiave. Per eliminare una KMS chiave, si utilizza l'ScheduleKeyDeletionoperazione. Dopo la

scadenza del periodo di attesa specificato, AWS KMS elimina la KMS chiave e registra una voce come la seguente nel CloudTrail registro per registrare l'evento.

CloudTrail le voci di registro relative a questa operazione registrate a partire da dicembre 2022 includono la chiave ARN della KMS chiave interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce la chiaveARN.

Per un esempio della voce di CloudTrail registro relativa all'`ScheduleKeyDeletion`operazione, vedereScheduleKeyDeletion. Per informazioni sull'eliminazione delle KMS chiavi, vedereEliminare un AWS KMS keys.

L'esempio seguente di voce di CloudTrail registro registra `DeleteKey` l'operazione di una KMS chiave con materiale chiave in 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 KMS chiave in un AWS CloudHSM archivio chiavi 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione `DescribeCustomKeyStores`. 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": {
        "customKeyStoreId": "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'DescribeKeyoperazione. AWS KMSregistra 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 gestione AWS 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": "2022-09-26T18:01:36Z",
    "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'DisableKeyoperazione. Per informazioni sull'attivazione e la disattivazione di AWS KMS keys in AWS KMS, consulta Attivazione e disattivazione dei tasti.

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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione DisableKeyRotation. 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione `DisconnectCustomKeyStore`. Per informazioni sulla disconnessione di un archivio delle chiavi personalizzate, consultare Disconnetti 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": "DisconnectCustomKeyStore",
    "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"
}
```

EnableKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'EnableKeyoperazione. Per informazioni sull'attivazione e la disattivazione di AWS KMS keys in AWS KMS, consulta Attivazione e disattivazione dei tasti.

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'EnableKeyRotationoperazione. Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene ruotata, vedeteRotateKey. Per informazioni sulla rotazione di AWS KMS keys, consulta Ruotare AWS KMS keys.

> **Note**
>
> rotation-periodÈ 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail per l'operazione Encrypt.

```
{
    "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": "2022-07-14T20:17:42Z",
    "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,
    "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",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'GenerateDataKeyoperazione.

```
{
    "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",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Department": "Engineering",
```

```
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "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

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'GenerateDataKeyPairoperazione. In questo esempio viene registrata un'operazione che genera una coppia di chiavi RSA crittografate con una AWS KMS key di crittografia simmetrica.

```
{
    "eventVersion": "1.05",
    "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,
    "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"
}
```

GenerateDataKeyPairWithoutPlaintext

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'GenerateDataKeyPairWithoutPlaintextoperazione. In questo esempio viene registrata un'operazione che genera una coppia di chiavi RSA crittografata con una AWS KMS key di crittografia simmetrica.

```
{
    "eventVersion": "1.05",
    "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": "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,
    "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"
}
```

GenerateDataKeyWithoutPlaintext

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'GenerateDataKeyWithoutPlaintextoperazione.

```
{
    "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": "2014-11-04T00:52:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "InvalidKeyUsageException",
```

```
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "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",
    "recipientAccountId": "111122223333"
}
```

GenerateMac

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'GenerateMacoperazione.

```
{
    "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'GenerateRandomoperazione. Poiché questa operazione non utilizza una AWS KMS key, il campo `resources` 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'GetKeyPolicyoperazione. 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'GetKeyRotationStatusoperazione. 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'GetParametersForImportoperazione. 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'ImportKeyMaterialoperazione. La stessa CloudTrail voce viene registrata quando si utilizza l'`ImportKeyMaterial`operazione o si utilizza la AWS KMS console per importare materiale chiave in unAWS 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": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-26T00:08:00Z",
    "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": "Jan 1, 2021 8:00:00 PM",
        "expirationModel": "KEY_MATERIAL_EXPIRES"
    },
    "responseElements": {
```

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

ListAliases

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'ListAliasesoperazione. Poiché questa operazione non utilizza alcun alias particolare o AWS KMS key, il campo `resources` è vuoto. Per informazioni sulla visualizzazione degli alias in AWS KMS, consulta 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"
    },
    "eventTime": "2014-11-04T00:51:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListAliases",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "limit": 5,
```

```
        "marker":
 "eyJiIjoiYWxpYXMvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwiYSI6ImFsaWFzL2U1NGNjMTkzL
    },
    "responseElements": null,
    "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ListGrants

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'ListGrantoperazione. Per informazioni sulle concessioni in AWS KMS, consulta 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":
 "eyJncmFudElkIjoiMWY4M2U2ZmM0YTY2NDgxYjQ2Yzc4MTdhM2Y4YmQwMDFkZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzc1N
\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'ListKeyRotationsoperazione. 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": "ListKeyRotations",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione PutKeyPolicy. Per informazioni sull'aggiornamento di una policy delle chiavi, consulta Modificare una politica 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'ReEncryptoperazione. Il campo `resources` in questa voce di log specifica due AWS KMS keys, la chiave KMS di origine e la chiave KMS di destinazione, in questo ordine.

```
{
    "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": "2020-07-27T23:09:13Z",
    "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,
    "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",
    "recipientAccountId": "111122223333"
}
```


ReplicateKey

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione ReplicateKey. Una `ReplicateKey` richiesta genera un'`ReplicateKey`operazione e un'CreateKeyoperazione.

Per informazioni sulla replica di chiavi in più Regioni, consulta Creazione di 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione RetireGrant. 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione RevokeGrant. 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 AWS KMS keys ruotano. Per informazioni sulla rotazione dei KMS tasti, vedere. Ruotare AWS KMS keys

L'esempio seguente mostra una voce di CloudTrail registro relativa all'operazione che ruota una KMS chiave di crittografia simmetrica sulla quale 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'`EnableKeyRotation`operazione, vedereEnableKeyRotation.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-14T01:41:59Z",
    "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",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "rotationType": "AUTOMATIC",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
```

```
}
```

L'esempio seguente mostra una voce di CloudTrail registro per un'RotateKeyOnDemandoperazione. Per informazioni sulla rotazione delle KMS chiavi di crittografia simmetriche su richiesta, vedere. Esegue la rotazione dei tasti su richiesta

Per un esempio della voce di CloudTrail registro che registra l'`RotateKeyOnDemand`operazione, vedere. RotateKeyOnDemand

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-14T01:41:59Z",
    "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",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "rotationType": "ON_DEMAND",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
}
```

RotateKeyOnDemand

L'esempio seguente mostra una voce di AWS CloudTrail registro per l'RotateKeyOnDemandoperazione. Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene ruotata, vedereRotateKey. Per ulteriori informazioni sulla rotazione su richiesta del materiale chiave per una chiave KMS, consulta. Esegue 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 AWS CloudTrail voci di registro relative all'ScheduleKeyDeletionoperazione.

Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene eliminata, vedereDeleteKey. Per informazioni sull'eliminazione AWS KMS keys, consulta Eliminare un AWS KMS keys.

L'esempio seguente registra una `ScheduleKeyDeletion` richiesta per una chiave Single-Region. 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": "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"
}
```

L'esempio seguente registra una `ScheduleKeyDeletion` richiesta per una chiave multiregionale con KMS chiavi di replica.

Perché 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 viene omesso.

```
{
    "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 di chiave in un KMS AWS CloudHSM archivio 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 log AWS CloudTrail 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-b125-4556-9702-12a2b9afcff7",
    "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 log AWS CloudTrail generata quando AWS KMS sincronizza una chiave per più Regioni. La sincronizzazione coinvolge le chiamate tra Regioni diverse per copiare le proprietà condivise di una chiave primaria in più Regioni alle relative chiavi di replica. AWS KMS

sincronizza periodicamente le chiavi in più Regioni per garantire che tutte le chiavi in più Regioni correlate abbiano lo stesso materiale chiave.

L'`resources`elemento della voce di CloudTrail registro include la chiave ARN della chiave primaria multiregionale, 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'TagResourceoperazione per aggiungere un tag con una chiave di tag `Department` e un valore di tag di`IT`.

Per un esempio di una voce di `UntagResource` CloudTrail registro che viene scritta quando la chiave viene ruotata, vedeteUntagResource. 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": "2020-07-01T21:19:25Z",
    "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'UntagResourceoperazione per eliminare un tag con una chiave di tag di`Dept`.

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'UpdateAliasoperazione. L'elemento `resources` include i campi per l'alias e le risorse della chiave KMS. Per informazioni sulla creazione di alias in AWS KMS, consulta 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

Nell'esempio seguente viene mostrata una voce di log AWS CloudTrail generata chiamando l'operazione `UpdateCustomKeyStore` per aggiornare l'ID cluster per un archivio chiavi personalizzate. 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

Nell'esempio seguente viene illustrata una voce di log AWS CloudTrail generata chiamando l'operazione UpdateKeyDescription.


```
{
    "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'UpdatePrimaryRegionoperazione 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 multiregionale 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'VerifyMacoperazione.


```
{
    "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 log AWS CloudTrail per l'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"
}
```

Esempio uno di Amazon EC2

L'esempio seguente registra la creazione di un volume crittografato da parte di un principale IAM utilizzando la chiave di volume predefinita nella console di gestione Amazon EC2.

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 gestione Amazon EC2. Il record del file di log di EC2 include un campo `volumeId` con un valore di `"vol-13439757"`. Il record AWS KMS contiene un campo `encryptionContext` 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"
    }
  ]
}
```

Esempio due di Amazon EC2

Nell'esempio seguente, un principale IAM che esegue un'istanza Amazon EC2 crea e monta un volume di dati crittografato in una chiave KMS. Questa azione genera più record di CloudTrail registro.

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 decrittare la chiave dati (`Decrypt`).

`instanceId` dell'istanza Amazon EC2, `"i-81e2f56c"`, viene visualizzato nell'evento `RunInstances`. 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 dei dati, `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, viene mostrato in tutte e tre le chiamate AWS KMS (`CreateGrant`, `GenerateDataKeyWithoutPlaintext` e `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": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
      },
      "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-bbc8-3d1b7cff49b0",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```


Monitora KMS le chiavi 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 vengono registrati per un periodo di due settimane in modo da poter accedere alle informazioni storiche e comprendere meglio l'utilizzo delle KMS chiavi e le loro modifiche nel tempo.

Puoi usare Amazon CloudWatch per avvisarti di eventi importanti, come i seguenti.

- Il materiale chiave importato in una KMS chiave si sta avvicinando alla data di scadenza.
- Una KMS chiave in attesa di eliminazione viene ancora utilizzata.

- Il materiale chiave contenuto in una KMS chiave è stato ruotato automaticamente.
- È stata eliminata una KMS chiave.

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 Gestire le tariffe delle AWS KMS API richieste 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 Amazon AWS Management Console e Amazon CloudWatch API.

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 Amazon AWS Management Console e Amazon CloudWatch API. Per ulteriori informazioni, consulta Visualizza i parametri disponibili nella Amazon CloudWatch User Guide.

Argomenti

- SecondsUntilKeyMaterialExpiration
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors

- XksExternalKeyManagerStates
- XksProxyLatency

SecondsUntilKeyMaterialExpiration

Il numero di secondi rimanenti alla scadenza del materiale chiave importato in una KMS chiave. Questa metrica è valida solo per KMS le chiavi con materiale chiave importato (l'origine del materiale chiave di`EXTERNAL`) e una data di scadenza.

Utilizza questo parametro per tenere traccia del tempo che rimane fino alla scadenza del materiale della chiave importato. Quando questo periodo di tempo scende al di sotto di una soglia definita, puoi reimportare il materiale della chiave con una nuova data di scadenza. La `SecondsUntilKeyMaterialExpiration` metrica è specifica per una KMS chiave. Non puoi utilizzare questa metrica per monitorare più KMS chiavi o KMS chiavi 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 KMS chiave.

Quando si pianifica l'eliminazione di una KMS chiave, AWS KMS impone un periodo di attesa prima di eliminare la KMS chiave. Puoi utilizzare il periodo di attesa per assicurarti di non aver bisogno della KMS chiave ora o in futuro. Puoi anche configurare un CloudWatch allarme per avvisarti se una persona o un'applicazione tenta di utilizzare la KMS chiave in un'operazione crittografica durante il periodo di attesa. Se ricevete una notifica da un allarme di questo tipo, potreste voler annullare l'eliminazione della KMS chiave.

Per istruzioni, consulta Crea un allarme che rileva l'uso di una KMS chiave in attesa di eliminazione.

ExternalKeyStoreThrottle

Il numero di richieste di operazioni crittografiche sulle KMS chiavi in ogni archivio di chiavi esterno che AWS KMS rallenta (risponde con un). `ThrottlingException` Questo parametro si applica solo agli archivi delle chiave esterne.

La `ExternalKeyStoreThrottle` metrica si applica solo alle KMS chiavi in un archivio di chiavi esterno e solo alle richieste per le operazioni crittografiche e l'operazione. DescribeKey AWS KMS limita queste richieste quando la frequenza delle richieste 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 questo parametro per verificare e modificare il valore della quota di richiesta dell'archivio delle chiavi personalizzate. Se questa metrica indica che AWS KMS spesso le richieste per queste KMS chiavi vengono limitate, 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 AWS Support e crea un caso.

Nome del gruppo di dimensioni: Keystore Throttle Metrics (Parametri di limitazione del keystore)

Dimensione	Descrizione
CustomKey StoreId	Valore per ogni archivio delle chiavi esterne.

Dimensione	Descrizione
KmsOperation	Valore per ogni operazione. AWS KMS API Questa metrica si applica solo alle operazioni crittografiche e alle `DescribeKey` operazioni sulle KMS chiavi in un archivio di chiavi esterno.
KeySpec	Valore per ogni tipo di chiave. KMS L'unica specifica chiave supportata per KMS le chiavi in un archivio di chiavi esterno è SYMMETRIC _DEFAULT.

XksProxyCertificateDaysToExpire

Il numero di giorni che mancano alla scadenza del TLS certificato per l'endpoint proxy dell'archivio chiavi esterno (`XksProxyUriEndpoint`). Questo parametro si applica solo agli archivi delle chiave esterne.

Usa questa metrica per creare un CloudWatch allarme che ti avvisi della scadenza imminente del certificato. TLS Quando il certificato scade, AWS KMS non è possibile comunicare con il proxy dell'archivio chiavi esterno. Tutti i dati protetti da KMS chiavi nell'archivio delle chiavi esterno 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: XKSProxy Certificate Metrics

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
CertificateName	Nome del soggetto (CN) nel TLS certificato.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi degli archivi di KMS chiavi esterni. Per istruzioni, consulta 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 chiave 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 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.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e KMS alle chiavi degli archivi di chiavi esterni. Per istruzioni, consulta 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 chiave esterne.

Utilizza questa metrica per tenere traccia del tasso di errore delle chiavi nel tuo archivio di chiavi esterno. KMS Rivela gli errori più frequenti, in modo da consentirti di assegnare priorità diverse agli interventi tecnici. Ad esempio, KMS le chiavi che generano tassi elevati di errori non ripetibili potrebbero indicare un problema con la configurazione 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.

Nome del gruppo di dimensioni: XKS Proxy Error Metrics

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
KmsOperation	Valore per ogni AWS KMS API operazione che ha generato una richiesta al XKS proxy.
XksOperation	Valore per ogni APIoperazione proxy di archiviazione delle chiavi esterne.
KeySpec	Valore per ogni tipo di KMS chiave. L'unica specifica chiave supportata per KMS le chiavi in un archivio di chiavi esterno 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
ExceptionName	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 degli archivi di KMS chiavi esterni. Per istruzioni, consulta 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 chiave 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 gestore di chiavi 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: External Key Manager Metrics XKS

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
XksExternalKeyManagerState	Valore per ogni stato di integrità.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi degli archivi di KMS chiavi esterni. Per istruzioni, consulta 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 chiave 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 tuo gestore di chiavi esterno non è in grado di gestire la frequenza di 1800 richieste al secondo, prendi in considerazione la possibilità di richiedere una riduzione della quota di richieste di KMS chiavi in un archivio di chiavi personalizzato. Le richieste di operazioni crittografiche che utilizzano le KMS chiavi dell'archivio di chiavi esterno falliranno rapidamente con un'eccezione di limitazione, anziché essere elaborate e successivamente rifiutate dal proxy dell'archivio chiavi esterno o dal gestore di chiavi esterno.

Nome del gruppo di dimensioni: XKS Proxy Latency Metrics

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
KmsOperation	Valore per ogni AWS KMS API operazione che ha generato una richiesta al XKS proxy.
XksOperation	Valore per ogni APIoperazione proxy di archiviazione delle chiavi esterne.
KeySpec	Valore per ogni tipo di KMS chiave. L'unica specifica chiave supportata per KMS le chiavi in un archivio di chiavi esterno è SYMMETRIC _DEFAULT.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi degli archivi di KMS chiavi esterni. Per istruzioni, consultare Monitora gli archivi di chiavi esterni.

Crea un CloudWatch allarme per la scadenza del materiale chiave importato

È possibile creare un CloudWatch allarme che avvisi quando il materiale chiave importato in una KMS chiave si avvicina alla scadenza. Ad esempio, l'allarme può avvisarti quando mancano meno di 30 giorni alla scadenza.

Quando importate materiale chiave in una KMS chiave, potete facoltativamente specificare una data e un'ora di scadenza del materiale chiave. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave e la chiave diventa inutilizzabile. KMS Per utilizzare nuovamente la KMS chiave,

è necessario reimportare il materiale chiave. Tuttavia, se si reimporta il materiale chiave prima che scada, è possibile evitare di interrompere i processi che utilizzano tale chiave. KMS

Questo allarme utilizza la `SecondsUntilKeyMaterialExpires`metrica AWS KMS pubblicata su CloudWatch per le KMS chiavi con materiale chiave importato che scade. Ogni allarme utilizza questa metrica per monitorare il materiale chiave importato per una chiave particolare. KMS Non è possibile creare un unico allarme per tutte le KMS chiavi con materiale chiave in scadenza o un allarme per KMS le chiavi che si potrebbero creare in futuro.

Requisiti

Le seguenti risorse sono necessarie per un CloudWatch allarme che monitora la scadenza del materiale chiave importato.

- Una KMS chiave con materiale chiave importato che scade.
- Un SNS argomento di Amazon. Per maggiori dettagli, consulta la sezione Creazione di un SNS argomento Amazon 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 Metriche per chiave. Scegli la riga con la KMS chiave e la metrica. `SecondsUntilKeyMaterialExpires` Quindi, scegli Seleziona parametro. L'elenco delle metriche visualizza la `SecondsUntilKeyMaterialExpires` metrica solo per KMS le chiavi con materiale chiave importato che scade. Se non disponi di KMS chiavi con queste proprietà nell'account e nella regione, questo elenco è vuoto.
Statistic	Minimo
Periodo	1 minuto

Campo	Valore
Tipo di soglia	Statico
Quando...	Ogni volta *metric-name* è maggiore di 1

Crea CloudWatch allarmi per archivi di chiavi esterni

Puoi creare CloudWatch allarmi Amazon basati su 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 (AmazonSNS) o a una politica di Amazon EC2 Auto 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 SNS argomento Amazon. Per maggiori dettagli, consulta la sezione Creazione di un SNS argomento Amazon 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 TLS certificato associato all'endpoint proxy dell'archivio di 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 scegli XKSProxy Certificate Metrics. Seleziona la casella di controllo accanto al valore `XksProxyCertificateName` da monitorare. Quindi, scegli Seleziona parametro.
Statistic	Minimo
Periodo	5 minuti
Tipo di soglia	Statico
Quando...	Ogni volta `Lower` che XksProxyCertificateDaysToExpireè 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 scegli XKSProxy Latency Metrics. Seleziona la casella di controllo accanto al valore `KmsOperation` da monitorare. Quindi, scegli Seleziona parametro.
Statistic	Media
Periodo	5 minuti
Tipo di soglia	Statico
Quando...	Ogni volta che XksProxyLatencyè così`Greater`. `200`

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).

Campo	Valore
	Inserisci `ErrorType = Retryable` in Filter by (Filtra per). Seleziona Esegui. Quindi, scegli Seleziona parametro.
Etichetta	*Retryable errors*
Periodo	1 minuto
Tipo di soglia	Statico
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.

Campo	Valore
Etichetta	*Non-retryable errors*
Periodo	1 minuto
Tipo di soglia	Statico
Quando...	Ogni volta che q1 è `Greater` di 5.

Monitora KMS le chiavi 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 contenuto in una KMS chiave veniva ruotato automaticamente.
- Il materiale chiave importato in una KMS chiave è scaduto.
- Una KMS chiave di cui era stata pianificata l'eliminazione è stata eliminata.

AWS KMS si integra con Amazon EventBridge per avvisarti di eventi importanti che influiscono sulle tue KMS chiavi. 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, SNS argomenti Amazon, Amazon SQS queues, stream 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 API richiesta di lettura/scrittura, consulta la Amazon EventBridge User Guide.

I seguenti argomenti descrivono gli EventBridge eventi che genera. AWS KMS

KMSCMKRotazione

AWS KMS supporta la rotazione automatica del materiale chiave in chiavi di crittografia KMS simmetriche. La rotazione annuale del materiale della chiave è facoltativa per le chiavi gestite dal cliente. Il materiale della chiave per le Chiavi gestite da AWS viene ruotato automaticamente ogni anno.

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": "2022-08-10T16:37:50Z",
  "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"
  }
}
```

KMSScadenza del materiale chiave importato

Quando importate materiale chiave in una KMS chiave, potete facoltativamente specificare l'ora in cui il materiale chiave scade. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave e invia un evento corrispondente a. `KMS Imported Key Material Expiration` 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": "2022-08-10T16:37:50Z",
  "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"
  }
}
```

KMSCMKEliminazione

Quando si pianifica l'eliminazione di una KMS chiave, AWS KMS impone un periodo di attesa prima di eliminare la KMS chiave. Al termine del periodo di attesa, AWS KMS elimina la KMS chiave e invia un `KMS CMK Deletion` evento a. EventBridge AWS KMS garantisce questo EventBridge evento. A causa di nuovi tentativi, potrebbe generare più eventi in pochi secondi che eliminano la stessa KMS chiave.

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": "2022-08-10T16:37:50Z",
  "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 consente di fare riferimento a una KMS chiave come `test-key` anziché a. `1234abcd-12ab-34cd-56ef-1234567890ab`

È possibile utilizzare un alias per identificare una KMS chiave nella AWS KMS console, nell'DescribeKeyoperazione e nelle operazioni crittografiche, ad esempio Encrypt e. GenerateDataKey Gli alias semplificano anche il riconoscimento delle Chiave gestita da AWS. Gli alias per queste KMS chiavi hanno sempre la stessa forma. `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 anche consentire e negare l'accesso alle KMS chiavi in base ai relativi alias senza modificare le politiche o gestire le concessioni. Questa funzionalità fa parte del AWS KMS supporto per il controllo degli accessi basato sugli attributi (). ABAC Per informazioni dettagliate, consultare Utilizzate gli alias per controllare l'accesso alle chiavi KMS.

Gran parte della potenza degli alias deriva dalla possibilità di modificare la KMS chiave associata a un alias in qualsiasi momento. Gli alias possono rendere il tuo codice più facile da scrivere e gestire. Ad esempio, supponiamo di utilizzare un alias per fare riferimento a una KMS chiave particolare e di voler modificare la chiave. KMS In tal caso, basta associare l'alias a una chiave diversa. 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 KMS chiave nella relativa regione. Quando il codice viene eseguito in ogni regione, l'alias fa riferimento alla KMS chiave associata in quella regione. Per vedere un esempio, consulta Scopri come utilizzare gli alias nelle tue applicazioni.

È possibile creare un alias per una KMS chiave nella AWS KMS console utilizzando o il CreateAliasAPImodello AWS::KMS: AWS CloudFormation :Alias.

AWS KMS APIFornisce il controllo completo degli alias in ogni account e regione. APIInclude operazioni per creare un alias (CreateAlias), visualizzare i nomi e gli alias ARNs (ListAliases) degli alias, modificare la KMS chiave 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 KMS chiave. Le azioni che esegui sull'alias non influiscono sulla chiave associataKMS. Puoi creare un alias per una KMS chiave e quindi aggiornare l'alias in modo che sia associato a una chiave diversa. KMS Puoi anche eliminare l'alias senza alcun effetto sulla chiave associata. KMS Tuttavia, se si elimina una KMS chiave, vengono eliminati tutti gli alias associati a tale KMS chiave.

Se si specifica un alias come risorsa in una IAM politica, la politica si riferisce all'alias, non alla chiave associata. KMS

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 alias ARN 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. In AWS KMS API, il nome dell'alias è sempre preceduto da. `alias/` Tale prefisso viene omesso 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 alla volta KMS

L'alias e la relativa KMS chiave 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 ListAliasesoutput mostra che l'`test-key`alias è associato esattamente a una KMS chiave di destinazione, rappresentata dalla `TargetKeyId` proprietà.

```
{
     "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
}
```

È possibile associare più alias alla stessa chiave KMS

Ad esempio, è possibile associare gli `project-key` alias `test-key` and alla stessa KMS chiave.

```
{
     "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 KMS chiave nella relativa 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 KMS chiave diversa. Per informazioni dettagliate, consultare Scopri come utilizzare gli alias nelle tue applicazioni.

È possibile modificare la KMS chiave associata a un alias

È possibile utilizzare l'UpdateAliasoperazione per associare un alias a una chiave diversaKMS. Ad esempio, se l'`finance-key`alias è associato alla `1234abcd-12ab-34cd-56ef-1234567890ab` KMS chiave, è possibile aggiornarlo in modo che sia associato alla `0987dcba-09fe-87dc-65ba-ab0987654321` KMS chiave.

Tuttavia, la KMS chiave corrente e quella nuova devono essere dello stesso tipo (entrambe simmetriche o entrambe asimmetriche o entrambeHMAC) e devono avere lo stesso utilizzo della chiave (ENCRYPT_ DECRYPT o _ o SIGN _ _). VERIFY 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 KMS chiave nella AWS KMS console, devi assegnarle un nuovo alias. Ma non è necessario un alias quando si utilizza l'CreateKeyoperazione per creare una KMS chiave. Inoltre, è possibile utilizzare l'UpdateAliasoperazione per modificare la KMS chiave associata a un alias e l'DeleteAliasoperazione per eliminare un alias. Di conseguenza, alcune KMS chiavi potrebbero avere diversi alias e altre potrebbero non averne nessuno.

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.

Utilizza gli alias per identificare le chiavi KMS

È possibile utilizzare un nome alias o un alias ARN per identificare una KMS chiave nelle operazioni crittografiche, e. DescribeKeyGetPublicKey (Se la KMSchiave si trova in un altro Account AWS, è necessario utilizzarne la chiave ARN o l'alias.) ARN Gli alias non sono identificatori validi per KMS le chiavi in altre operazioni. AWS KMS Per informazioni sugli identificatori di chiave validi per ogni AWS KMS API operazione, vedete le descrizioni dei *`KeyId`* parametri nel riferimento.AWS Key Management Service API

Non è possibile utilizzare un nome alias o un alias ARN per identificare una KMS chiave in una politica. IAM Per controllare l'accesso a una KMS chiave in base ai relativi alias, usa le chiavi di condizione kms: RequestAlias o kms:. ResourceAliases Per informazioni dettagliate, consultare ABACper AWS KMS.

Controllo dell'accesso agli alias

Quando crei o modifichi un alias, influisci sull'alias e sulla chiave associata. KMS Pertanto, i responsabili che gestiscono gli alias devono essere autorizzati a richiamare l'operazione alias sull'alias e su tutte le chiavi interessate. KMS È possibile fornire queste autorizzazioni utilizzando politiche, politiche e concessioni chiave. IAM

> **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 ABACper AWS KMS e Utilizzate gli alias per controllare l'accesso alle chiavi KMS.

Per informazioni sul controllo dell'accesso a tutte le AWS KMS operazioni, vedere. Riferimento per le autorizzazioni

Le autorizzazioni per la creazione e la gestione degli alias funzionano come descritto di seguito.

km: CreateAlias

Per creare un alias, il principale necessita delle seguenti autorizzazioni sia per l'alias che per la chiave associata. KMS

- `kms:CreateAlias` per l'alias. Fornisci questa autorizzazione in una IAM politica allegata al principale autorizzato a creare l'alias.

  L'esempio di istruzione di policy seguente specifica un alias particolare in un elemento `Resource`. 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. 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 politica chiave o in una IAM politica delegata dalla politica chiave.

```
{
  "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 condizionali per limitare le KMS chiavi 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 asimmetriche. KMS Per un elenco completo delle chiavi di condizione che puoi utilizzare per limitare l'`kms:CreateAlias`autorizzazione sulle KMS risorse chiave, consulta. AWS KMS autorizzazioni

km: ListAliases

Per elencare gli alias nell'account e nella regione, il principale deve disporre dell'`kms:ListAliases`autorizzazione in una IAM politica. Poiché questa politica non è correlata a nessuna KMS chiave o risorsa alias particolare, il valore dell'elemento risorsa nella politica deve essere. `"*"`

Ad esempio, la seguente dichiarazione IAM politica fornisce l'autorizzazione principale per elencare tutte le KMS chiavi e gli alias nell'account e nella regione.

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

```
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

km: UpdateAlias

Per modificare la KMS chiave associata a un alias, il principale necessita di tre elementi di autorizzazione: uno per l'alias, uno per la KMS chiave corrente e uno per la nuova chiave. KMS

Ad esempio, supponiamo di voler modificare l'`test-key`alias dalla KMS chiave con ID chiave 1234abcd-12ab-34cd-56ef-1234567890ab alla KMS chiave 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 IAM viene fornita in una politica allegata al principale. La seguente IAM politica 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 KMS chiave attualmente associata all'alias. Questa autorizzazione deve essere fornita in una politica chiave o in una IAM politica delegata dalla politica chiave.

  ```
  {
    "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 KMS chiave che l'operazione associa all'alias. Questa autorizzazione deve essere fornita in una politica chiave o in una IAM politica delegata dalla politica chiave.

```
{
  "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 i tasti condizionali per limitare una o entrambe le KMS chiavi in un'`UpdateAlias`operazione. Ad esempio, puoi utilizzare una chiave kms: ResourceAliases condition per consentire al principale di aggiornare gli alias solo quando la KMS chiave di destinazione ha già un alias particolare. Per un elenco completo delle chiavi di condizione che puoi utilizzare per limitare l'`kms:UpdateAlias`autorizzazione su una risorsa KMS chiave, vedi. AWS KMS autorizzazioni

km: DeleteAlias

Per eliminare un alias, il principale necessita dell'autorizzazione per l'alias e per la chiave associata. KMS

Come sempre, è necessario prestare attenzione quando si concedono le autorizzazioni per eliminare una risorsa. Tuttavia, l'eliminazione di un alias non ha alcun effetto sulla chiave associata. KMS 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 IAM politica allegata al principale che è autorizzato a eliminare l'alias.

L'esempio di istruzione di policy seguente specifica l'alias in un elemento `Resource`. È tuttavia possibile elencare più alias ARNs o specificare un modello di alias, ad esempio`"test*"`, È possibile 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 associata. KMS Questa autorizzazione deve essere fornita in una politica chiave o in una IAM politica delegata dalla politica chiave.

```
{
  "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 condizionali per limitare i permessi degli alias quando la risorsa è una chiave. KMS Ad esempio, la seguente IAM politica consente le operazioni di alias sulle KMS chiavi in un account e in una regione particolari. Tuttavia, utilizza la chiave kms: KeyOrigin condition per

limitare ulteriormente le autorizzazioni alle KMS chiavi con materiale chiave proveniente da. AWS KMS

Per un elenco completo delle chiavi di condizione che puoi utilizzare per limitare l'autorizzazione degli alias su una risorsa KMS chiave, 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, utilizzate il valore dell'`Resource`elemento della dichiarazione IAM politica che controlla l'accesso all'alias. Ad esempio, le seguenti istruzioni 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 AWS KMS API le operazioni.

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 KMS chiave o per una chiave esistenteKMS. È possibile aggiungere un alias in modo che una determinata KMS chiave venga utilizzata in un progetto o in un'applicazione.

È inoltre possibile utilizzare un AWS CloudFormation modello per creare un alias per una KMS chiave. Per ulteriori informazioni, consulta AWS::KMS: :Alias nella Guida per l'AWS CloudFormation utente.

Utilizzo della console AWS KMS

Quando si crea una KMS chiave nella AWS KMS console, è necessario creare un alias per la nuova KMS chiave. Per creare un alias per una KMS chiave esistente, utilizza la scheda Alias nella pagina di dettaglio della chiave. KMS

1. Accedi a AWS Management Console 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 gestire gli alias per o. Chiavi gestite da AWS Chiavi di proprietà di AWS
4. Nella tabella, scegli l'ID o l'alias della KMS chiave. Quindi, nella pagina dei dettagli della KMS chiave, scegli la scheda Alias.

Se una KMS chiave ha più alias, la colonna Alias della tabella visualizza un alias e un riepilogo degli alias, ad esempio (+ n more). Scegliendo il riepilogo degli alias si accede direttamente alla scheda Alias nella pagina di dettaglio 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`.

Usando il AWS KMS API

Per creare un alias, utilizzare l'CreateAliasoperazione. A differenza del processo di creazione delle KMS chiavi nella console, l'CreateKeyoperazione non crea un alias per una nuova KMS chiave.

> **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.

È possibile utilizzare l'`CreateAlias`operazione per creare un alias per una nuova KMS chiave senza alias. È inoltre possibile utilizzare l'`CreateAlias`operazione per aggiungere un alias a qualsiasi KMS chiave esistente o per ricreare un alias eliminato accidentalmente.

Nelle AWS KMS API operazioni, 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'ListAliasesoperazione. 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 KMS chiave, utilizzate il relativo ID o la chiave ARN. Non puoi utilizzare un altro alias.

L'esempio seguente crea l'`example-key`alias e lo associa alla chiave specificataKMS. Questi esempi utilizzano il AWS Command Line Interface ()AWS CLI. Per esempi in più linguaggi di programmazione, consulta Utilizzare `CreateAlias` con un AWS SDK o 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, consultare Usando il AWS KMS API.

Trova il nome dell'alias e l'alias ARN per una chiave KMS

Gli alias facilitano il riconoscimento KMS delle chiavi nella AWS KMS console. È possibile visualizzare gli alias di una KMS chiave nella AWS KMS console o utilizzando l'ListAliasesoperazione. L'DescribeKeyoperazione, che restituisce le proprietà di una KMS chiave, non include gli alias.

Le procedure seguenti mostrano come visualizzare e identificare gli alias associati a una KMS chiave utilizzando la AWS KMS console e. AWS KMS API Negli AWS KMS API esempi viene utilizzato il AWS Command Line Interface (AWS CLI), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

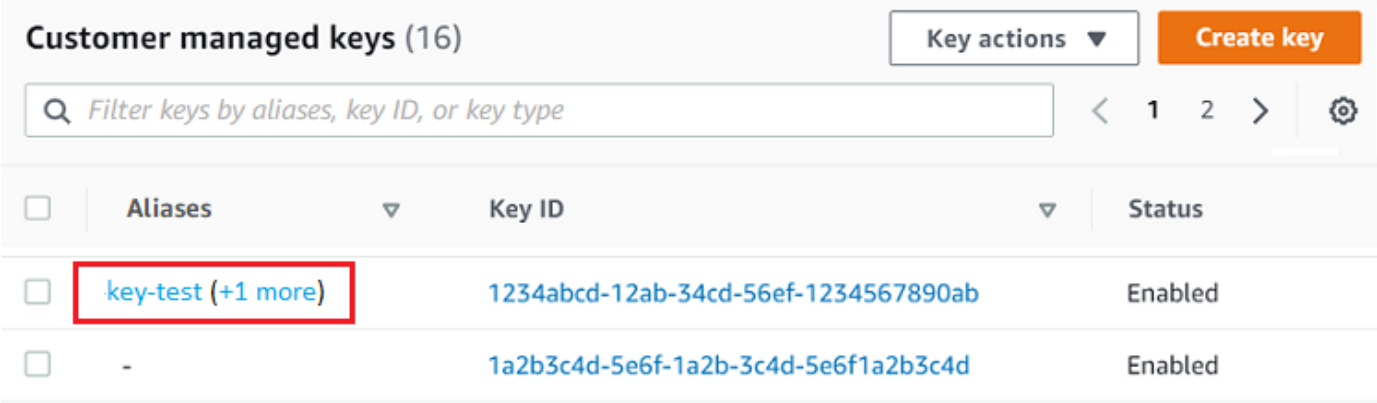
La AWS KMS console visualizza gli alias associati alla KMS chiave.

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. La colonna Alias mostra l'alias per ogni KMS chiave. Se una KMS chiave non ha un alias, nella colonna Alias viene visualizzato un trattino (-).

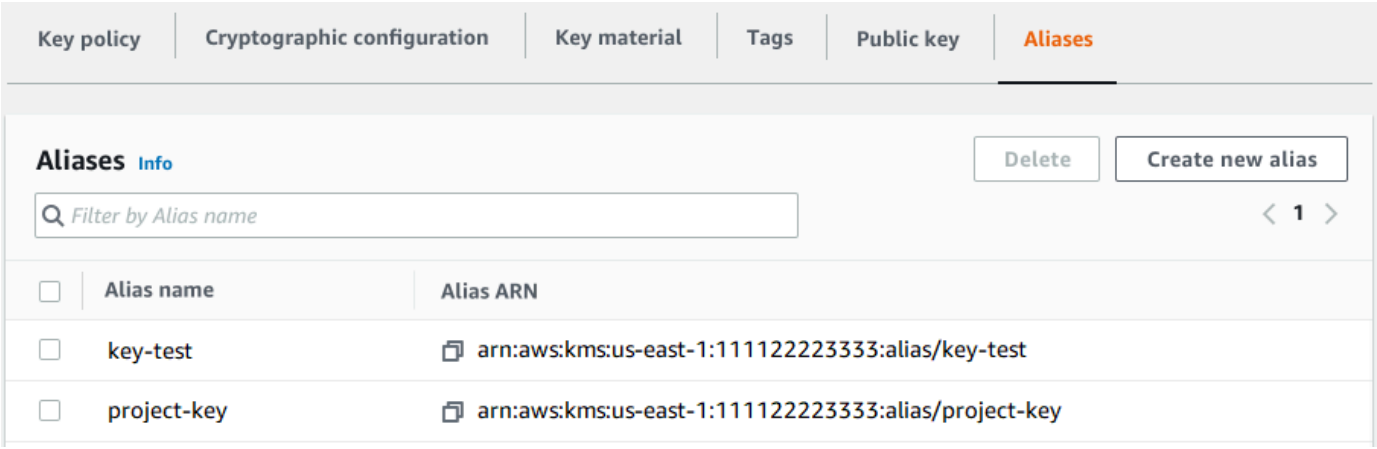
Se una KMS chiave ha più alias, la colonna Alias contiene anche un riepilogo degli alias, ad esempio (+ n more). Ad esempio, la KMS chiave seguente ha due alias, uno dei quali è. `key-test`

Per trovare il nome alias e l'alias ARN di tutti gli alias della KMS chiave, utilizzate la scheda Alias.

- Per andare direttamente alla scheda Alias, nella colonna Alias, scegli il riepilogo degli alias (+n più). Un riepilogo degli alias viene visualizzato solo se la KMS chiave ha più di un alias.
- In alternativa, scegli l'alias o l'ID della KMS chiave (che apre la pagina di dettaglio della KMS chiave), quindi scegli la scheda Alias. Le schede si trovano sotto la sezione Configurazione generale.

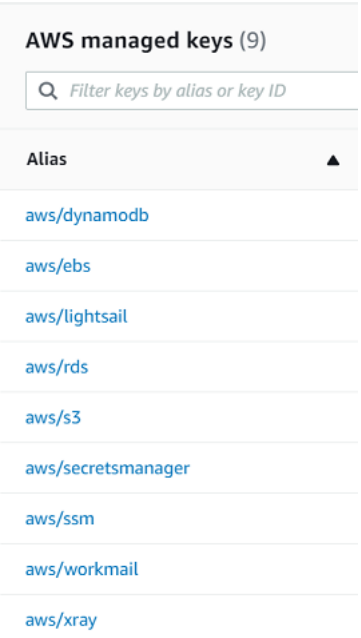


5. La scheda Alias mostra il nome alias e l'alias di tutti gli alias ARN di una chiave. KMS In questa scheda è inoltre possibile creare ed eliminare alias per la KMS chiave.



Chiavi gestite da AWS

È possibile utilizzare l'alias per riconoscere un Chiave gestita da AWS, come mostrato in questa pagina di esempio Chiavi gestite da AWS. 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`



Usando il AWS KMS API

L'ListAliasesoperazione restituisce il nome 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 KMS chiave, utilizzate il `KeyId` parametro opzionale dell'operazione. `ListAliases` Il `KeyId` parametro accetta l'ID o la chiave ARN della KMS chiave.

Questo esempio ottiene tutti gli alias associati alla `0987dcba-09fe-87dc-65ba-ab0987654321` KMS chiave.

```
$ 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, è possibile modificare la KMS chiave associata a un alias. Ad esempio, se l'`test-key`alias è associato a una KMS chiave, è possibile utilizzare l'UpdateAliasoperazione per associarlo a una chiave diversa. KMS Questo è uno dei diversi modi per ruotare manualmente una KMS chiave senza modificarne il materiale. È inoltre possibile aggiornare una KMS chiave in modo che un'applicazione che utilizzava una KMS chiave per nuove risorse ora utilizzi una KMS chiave diversa.

Non è possibile aggiornare un alias nella AWS KMS console. Inoltre, non puoi utilizzare `UpdateAlias` (o qualsiasi altra operazione) per modificare un nome di alias. Per modificare il nome di un alias, elimina l'alias corrente e quindi crea un nuovo alias per la chiave. KMS

Quando aggiorni un alias, la KMS chiave corrente e la nuova KMS chiave devono essere dello stesso tipo (simmetrica o asimmetrica o). HMAC Devono inoltre avere lo stesso utilizzo della chiave (`ENCRYPT_DECRYPT`o _ _). `SIGN_VERIFY` GENERATE VERIFY MAC Questa restrizione impedisce errori di crittografia nel codice che utilizza alias.

L'esempio seguente inizia utilizzando l'ListAliasesoperazione per mostrare che l'`test-key`alias è attualmente associato alla KMS chiave`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'`UpdateAlias`operazione per modificare la KMS chiave associata all'`test-key`alias in key. KMS `0987dcba-09fe-87dc-65ba-ab0987654321` Non è necessario specificare la chiave attualmente associata, ma solo la nuova KMS chiave («target»)KMS. 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 KMS chiave di destinazione, usa nuovamente l'`ListAliases`operazione. Questo AWS CLI comando utilizza il `--query` parametro per ottenere solo l'`test-key`alias. 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'DeleteAliasoperazione. Prima di eliminare un alias, assicurati che non sia in uso. Sebbene l'eliminazione di un alias non influisca sulla KMS chiave associata, potrebbe creare problemi a qualsiasi applicazione che utilizza l'alias. Se si elimina un alias per errore, è possibile creare un nuovo alias con lo stesso nome e associarlo alla stessa chiave o a una chiave diversa. KMS

Se si elimina una KMS chiave, vengono eliminati tutti gli alias associati a tale KMS chiave.

Utilizzo della console AWS KMS

Per eliminare un alias nella AWS KMS console, utilizza la scheda Alias nella pagina di dettaglio della chiave. KMS È possibile eliminare più alias per una KMS chiave contemporaneamente.

1. Accedi AWS Management Console 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 gli alias per o. Chiavi gestite da AWS Chiavi di proprietà di AWS

4. Nella tabella, scegli l'ID o l'alias della KMS chiave. Quindi, nella pagina dei dettagli della KMS chiave, scegli la scheda Alias.

   Se una KMS chiave ha più alias, la colonna Alias della tabella visualizza un alias e un riepilogo degli alias, ad esempio (+ n more). Scegliendo il riepilogo degli alias si accede direttamente alla scheda Alias nella pagina di dettaglio chiave. KMS
5. Sulla scheda Alias seleziona la casella di controllo accanto agli alias che si desidera eliminare. Scegli Elimina.

Utilizzando il AWS KMS API

Per eliminare un alias, utilizzare l'DeleteAliasoperazione. 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'ListAliasesoperazione. 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`]'
[]
```

Utilizzate gli alias per controllare l'accesso alle chiavi KMS

È possibile controllare l'accesso alle KMS chiavi in base agli alias associati alla KMS chiave. Per farlo, usa i tasti kms: RequestAlias e kms: ResourceAliases condition. Questa funzionalità fa parte del AWS KMS supporto per il controllo degli accessi basato sugli attributi (). ABAC

La chiave di `kms:RequestAlias` condizione consente o nega l'accesso a una KMS chiave in base all'alias contenuto in una richiesta. La chiave `kms:ResourceAliases` condizionale consente o nega l'accesso a una KMS chiave in base agli alias associati alla chiave. KMS

Queste funzionalità non consentono di identificare una KMS chiave utilizzando un alias nell'`resource`elemento di una dichiarazione politica. Quando un alias è il valore di un `resource` elemento, la policy si applica alla risorsa alias, non a qualsiasi KMS chiave che potrebbe essere associata ad essa.

> **Note**
>
> Potrebbero essere necessari fino a cinque minuti prima che le modifiche ai tag e agli alias influiscano sull'KMSautorizzazione delle chiavi. Le modifiche recenti potrebbero essere visibili nelle API operazioni prima che influiscano sull'autorizzazione.

Quando utilizzi gli alias per controllare l'accesso alle KMS chiavi, considera quanto segue:

- Utilizzare gli alias per rafforzare la best practice dell'accesso con privilegio minimo. IAMAssegna ai responsabili solo le autorizzazioni di cui hanno bisogno solo per le KMS chiavi che devono usare o gestire. Ad esempio, utilizzate gli alias per identificare le KMS chiavi utilizzate per un progetto. Quindi concedi al team di progetto il permesso di utilizzare solo KMS le chiavi 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 usi gli alias per controllare l'accesso alle KMS chiavi, la modifica di un alias può dare ai responsabili il permesso di usare KMS chiavi che altrimenti non avrebbero avuto il permesso di usare. Può anche negare l'accesso alle KMS chiavi di cui altri dirigenti hanno bisogno 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 chiave che non sono autorizzati a modificare le politiche chiave o a creare sovvenzioni possono controllare l'accesso alle KMS chiavi se sono autorizzati a 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`. IAMle politiche potrebbero concedere autorizzazioni di alias per tutte le chiavi del tuo. KMS Account AWS Ad esempio, la policy AWSKeyManagementServicePowerUsergestita consente ai principali di creare, eliminare ed elencare alias per tutte le KMS chiavi ma non di aggiornarle.
- Prima di impostare una politica che dipenda da un alias, esamina gli alias delle chiavi del KMS tuo. Account AWS Assicurati che la policy sia valida solo per gli alias che intendi includere. Usa

CloudTrail registri e CloudWatch allarmi per avvisarti delle modifiche agli alias che potrebbero influire sull'accesso alle tue chiavi. KMS Inoltre, la ListAliasesrisposta 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 `kms:ResourceAliases` condizione dipende dagli alias associati a una KMS chiave, anche se non compaiono nella richiesta.

km: RequestAlias

Consenti o nega l'accesso a una KMS chiave in base all'alias che identifica la KMS chiave in una richiesta. È possibile utilizzare la chiave kms: RequestAlias condition in una politica o in una politica chiave. IAM Si applica alle operazioni che utilizzano un alias per identificare una KMS chiave in una richiesta, vale a dire le operazioni crittografiche, DescribeKeye. GetPublicKey Non è valido per le operazioni di alias, come o. CreateAliasDeleteAlias

Nella chiave condizione, specificare un Nome alias o modello di nome alias. Non è possibile specificare un alias ARN.

Ad esempio, la seguente dichiarazione di politica chiave consente ai principali di utilizzare le operazioni specificate sulla KMS chiave. L'autorizzazione è effettiva solo quando la richiesta utilizza un alias che include `alpha` l'identificazione della KMS chiave.

```
{
  "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 utilizza un ID chiave, una chiave ARN o un alias diverso non soddisferebbe la condizione, anche se questi valori identificassero la stessa KMS chiave.

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-
alpha"
```

km: ResourceAliases

Consenti o nega l'accesso a una KMS chiave in base agli alias associati alla KMS chiave, 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 IAM politica per controllare l'accesso a più KMS chiavi nella stessa `alias/test*` regione. È valido per qualsiasi AWS KMS operazione che utilizza una chiave. KMS

Ad esempio, la seguente IAM politica consente ai principali di richiamare in due Account AWS le operazioni specificate sulle KMS chiavi. Tuttavia, l'autorizzazione si applica solo alle KMS chiavi associate agli alias che iniziano con. `restricted`

```
{
  "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, anche la seguente richiesta di esempio soddisfa la condizione, a condizione che la KMS chiave specificata abbia un alias che inizia con`restricted`, anche se tale 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 usare un alias per rappresentare una KMS chiave nel codice dell'applicazione. Il `KeyId` parametro è utilizzato nelle operazioni AWS KMS crittografiche e GetPublicKeyaccetta un nome alias o un alias. DescribeKeyARN

Ad esempio, il `GenerateDataKey` comando seguente utilizza un alias name (`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 KMS chiave si trova in un altro Account AWS, è necessario utilizzare una chiave ARN o un alias ARN in queste operazioni. Quando usi un aliasARN, ricorda che l'alias di una KMS chiave è definito nell'account che possiede la KMS chiave e potrebbe differire in ogni regione. Per informazioni su come trovare l'alias, consultaARN. Trova il nome dell'alias e l'alias ARN per una chiave KMS

Ad esempio, il `GenerateDataKey` comando seguente utilizza una KMS chiave non presente nell'account del chiamante. L'`ExampleAlias`alias è associato alla KMS chiave 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, potresti avere un'applicazione globale che utilizza una KMSchiave RSA asimmetrica 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`.

È possibile creare una versione diversa dell'applicazione in ogni regione o utilizzare un dizionario o un'istruzione switch per selezionare la KMS chiave giusta 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
```

Quindi, utilizza l'alias nel codice. Quando il codice viene eseguito in ogni regione, l'alias farà riferimento alla KMS chiave 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 venga eliminato o aggiornato per essere associato a una chiave diversaKMS. 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, consultare Controllo dell'accesso agli alias.

Esistono diverse altre soluzioni per le applicazioni che crittografano i dati in più regioni 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'operazione. AWS KMS API Quando lo fate, l'alias e la chiave ARN della KMS chiave vengono registrati nella voce di AWS CloudTrail registro dell'evento. L'alias viene visualizzato nel campo `requestParameters`. La chiave ARN appare nel `resources` campo. Questo è vero anche quando un AWS servizio ne utilizza uno Chiave gestita da AWS nel tuo account.

Ad esempio, la GenerateDataKeyrichiesta seguente utilizza l'`project-key`alias per rappresentare una KMS chiave.

```
$ 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 KMS chiave 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 botocore/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 AWS KMS API chiamate 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 al momento della creazione della KMS chiave e taggare o rimuovere tag alle KMS chiavi esistenti a meno che non siano in attesa di eliminazione. Non è possibile contrassegnare alias, Chiavi gestite da AWS archivi di chiavi personalizzati o KMS chiavi in Chiavi di proprietà di AWS altri archivi. Account AWS I tag sono opzionali, ma possono essere molto utili.

Ad esempio, puoi aggiungere un `"Project"="Alpha"` tag a tutte le KMS chiavi e ai bucket Amazon S3 che usi 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 KMS chiave e a un volume o AWS Secrets Manager segreto Amazon Elastic Block Store (AmazonEBS). Puoi anche utilizzare i tag per identificare KMS le chiavi per l'automazione.
- Tieni traccia AWS dei costi. 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 KMS chiavi 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. Utilizzate i tag per controllare l'accesso alle KMS chiavi Per informazioni più generali sull'uso dei tag per controllare l'accesso alle AWS risorse, consulta Controllare l'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida per l'IAMutente.

AWS KMS scrive una voce nel AWS CloudTrail registro quando si utilizzano le ListResourceTagsoperazioni TagResourceUntagResource, o.

Argomenti

- Controllo degli accessi ai tag
- Aggiungere tag a una KMS chiave
- Modifica i tag associati a una KMS chiave
- Rimuovi i tag associati a una KMS chiave
- Visualizza i tag associati a una KMS chiave
- Utilizzate i tag per controllare l'accesso alle KMS chiavi

Controllo degli accessi ai tag

Per aggiungere, visualizzare ed eliminare i tag, nella AWS KMS console o utilizzando, i principali devono disporre delle API autorizzazioni di etichettatura. Puoi fornire queste autorizzazioni nelle policy delle chiavi. Puoi anche fornirli nelle IAM policy (incluse le policy VPCdegli endpoint), ma solo se la policy chiave lo consente. La policy AWSKeyManagementServicePowerUsergestita consente ai responsabili di etichettare, rimuovere i tag ed elencare i tag su tutte le KMS chiavi a cui l'account può accedere.

Puoi anche limitare queste autorizzazioni utilizzando 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 ABACper AWS KMS e Utilizzate i tag per controllare l'accesso alle KMS chiavi.

Per esempio, politiche e ulteriori informazioni, vedere Controlling Access Based on Tag Keys nella Guida per l'IAMutente.

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 KMS chiave, il principale deve disporre dell'autorizzazione in una IAM politica che non sia limitata a KMS chiavi particolari.

kms:ListResourceTags

Consente ai principali di visualizzare i tag sulle KMS chiavi.

kms:UntagResource

Consente ai principali di eliminare i tag dalle KMS chiavi.

Autorizzazioni ad assegnare tag nelle policy

È possibile fornire le autorizzazioni di etichettatura in una politica o in una politica chiave. IAM Ad esempio, la politica chiave di esempio seguente fornisce a determinati utenti l'autorizzazione a contrassegnare la chiave. KMS Fornisce a tutti gli utenti che possono assumere l'esempio dei ruoli di Amministratore o Sviluppatore il permesso di visualizzare i tag.

```
{
  "Version": "2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
```

```
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow all tagging permissions",
      "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": "Allow roles to view tags",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:role/Administrator",
        "arn:aws:iam::111122223333:role/Developer"
      ]},
      "Action": "kms:ListResourceTags",
      "Resource": "*"
    }
  ]
}
```

Per concedere ai principali il permesso di etichettare più KMS chiavi, puoi utilizzare una policy. IAM Affinché questa politica sia efficace, la politica chiave per ogni KMS chiave deve consentire all'account di utilizzare IAM le politiche per controllare l'accesso alla chiave. KMS

Ad esempio, la seguente IAM politica consente ai principali di creare KMS chiavi. Consente inoltre loro di creare e gestire tag su tutte le KMS chiavi dell'account specificato. Questa combinazione consente ai responsabili di utilizzare il parametro Tags dell'CreateKeyoperazione per aggiungere tag a una KMS chiave durante la creazione.

```
{
  "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 `kms:KeyOrigin` condizione per impedire ai principali di etichettare o decontrassegnare KMS le chiavi con materiale chiave importato.

- leggi: RequestTag
- aws:ResourceTag/tag-key (solo IAM politiche)
- è stato: TagKeys
- km: CallerAccount
- km: KeySpec
- km: KeyUsage
- km: KeyOrigin
- km: ViaService

Come best practice, quando usi i tag per controllare l'accesso alle KMS chiavi, usa il tasto `aws:RequestTag/tag-key` o `aws:TagKeys` condition per determinare quali tag (o chiavi di tag) sono consentiti.

Ad esempio, la seguente IAM politica 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` o `ForAnyValue` impostare 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 i dettagli sugli operatori impostati, consulta Utilizzare più chiavi e valori nella Guida per l'IAMutente.

```
{
  "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"}
```

```
      }
    }
  ]
}
```

Aggiungere tag a una KMS chiave

I tag aiutano a identificare e organizzare le AWS risorse. Puoi aggiungere tag a una chiave gestita dal cliente quando crei la KMS chiave o aggiungere tag a KMS chiavi 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 AWS KMS API. Negli AWS KMS API esempi viene utilizzato il AWS Command Line Interface (AWS CLI), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Argomenti

- Aggiungi tag durante la creazione di una KMS chiave
- Aggiungi tag alle KMS chiavi esistenti

Aggiungi tag durante la creazione di una KMS chiave

È possibile aggiungere tag a una KMS chiave mentre si crea la chiave utilizzando la AWS KMS console o l'CreateKeyoperazione. Per aggiungere tag durante la creazione di una KMS chiave, è necessario disporre `kms:TagResource` dell'autorizzazione in una IAM politica oltre alle autorizzazioni necessarie per creare KMS le chiavi. Come minimo, l'autorizzazione deve coprire tutte le KMS chiavi dell'account e della regione. Per informazioni dettagliate, consultare Controllo degli accessi ai tag .

Utilizzo della AWS KMS console

Per aggiungere tag quando si crea una KMS chiave nella console, è necessario disporre delle autorizzazioni necessarie per visualizzare KMS le chiavi nella console oltre alle autorizzazioni necessarie per etichettare e creare KMS chiavi. Come minimo, l'autorizzazione deve coprire tutte le KMS chiavi dell'account e della regione.

1. Accedi AWS Management Console 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 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). Quando hai finito di etichettare la tua nuova KMS chiave, scegli Avanti.
7. Termina la creazione della tua KMS chiave.

Usando il AWS KMS API

Per specificare i tag durante la creazione di chiavi utilizzando l'CreateKeyoperazione, 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 su una KMS chiave 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 KMS chiave di crittografia simmetrica con un `Project:Alpha` tag. 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 `KeyMetadata` oggetto con informazioni sulla nuova KMS chiave. Tuttavia, `KeyMetadata` non include tag. Per ottenere i tag, usa l'ListResourceTagsoperazione.

Aggiungi tag alle KMS chiavi esistenti

Puoi aggiungere tag alle KMS chiavi esistenti gestite dai clienti nella AWS KMS console o utilizzando l'TagResourceoperazione. Per aggiungere tag, è necessario il permesso di etichettare la KMS chiave. È possibile ottenere questa autorizzazione dalla politica chiave per la KMS chiave o, se la politica chiave lo consente, da una IAM politica che include la KMS chiave.

Utilizzo della AWS KMS console

1. Accedi AWS Management Console 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. Puoi usare il filtro della tabella per visualizzare solo le KMS chiavi con tag particolari. Per i dettagli, consulta Visualizzare i tag utilizzando la AWS KMS console.
5. Seleziona la casella di controllo accanto all'alias di una KMS chiave.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della KMS chiave, 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).

Usando il AWS KMS API

L'TagResourceoperazione aggiunge uno o più tag a una KMS chiave. 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 dei tag”.

Per aggiungere un tag, specifica una nuova chiave di tag e un valore di tag. Ogni tag su una KMS chiave deve avere una chiave di tag diversa. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, il comando seguente aggiunge **`Purpose Department`** tag a una KMS chiave 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 KMS chiave, utilizzate l'ListResourceTagsoperazione.

Modifica i tag associati a una KMS chiave

I tag aiutano a identificare e organizzare le AWS risorse. Puoi modificare i tag associati alle KMS chiavi gestite dai clienti nella AWS KMS console o utilizzando l'TagResourceoperazione. Non è possibile modificare i tag di un Chiave gestita da AWS.

Le seguenti procedure mostrano come modificare i tag associati a una KMS chiave. Negli AWS KMS API esempi viene utilizzato il AWS Command Line Interface (AWS CLI), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi AWS Management Console 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 della tabella per visualizzare solo KMS le chiavi con tag particolari. Per i dettagli, consulta Visualizzare i tag utilizzando la AWS KMS console.
5. Seleziona la casella di controllo accanto all'alias di una KMS chiave.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della KMS chiave, 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).

Usando il AWS KMS API

L'TagResourceoperazione aggiunge uno o più tag a una chiave gestita dal cliente;. Tuttavia, puoi anche usare TagResourceper 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 su una KMS chiave deve avere una chiave di tag diversa. 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 KMS chiave

I tag aiutano a identificare e organizzare le AWS risorse. Puoi rimuovere i tag associati alle KMS chiavi gestite dai clienti nella AWS KMS console o utilizzando l'UntagResourceoperazione. Non è possibile modificare o rimuovere i tag di un Chiave gestita da AWS.

Le seguenti procedure mostrano come rimuovere i tag da una KMS chiave. Negli AWS KMS API esempi viene utilizzato il AWS Command Line Interface comando (AWS CLI), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi AWS Management Console 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. Puoi usare il filtro della tabella per visualizzare solo le KMS chiavi con tag particolari. Per i dettagli, consulta Visualizzare i tag utilizzando la AWS KMS console.
5. Seleziona la casella di controllo accanto all'alias di una KMS chiave.

6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della KMS chiave, 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).

Usando il AWS KMS API

L'UntagResourceoperazione elimina i tag da una KMS chiave. Per identificare i tag da eliminare, specifica le chiavi dei tag. Non è possibile utilizzare questa operazione per eliminare tag da KMS chiavi diverse Account AWS.

Quando l'operazione `UntagResource` ha esito positivo non restituisce alcun output. Inoltre, se la chiave del tag specificata non viene trovata sulla KMS chiave, non genera un'eccezione né restituisce una risposta. Per confermare che l'operazione ha funzionato, usa l'ListResourceTagsoperazione.

Ad esempio, questo comando elimina il **`Purpose`** tag e il relativo valore dalla KMS chiave specificata.

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys
 Purpose
```

Visualizza i tag associati a una KMS chiave

I tag aiutano a identificare e organizzare AWS le tue risorse. È possibile visualizzare i tag associati alle KMS chiavi gestite dai clienti nella AWS KMS console o utilizzando l'ListResourceTagsoperazione.

Le seguenti procedure mostrano come trovare i tag associati a una KMS chiave specifica. Negli AWS KMS API esempi viene utilizzato il AWS Command Line Interface (AWS CLI), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi AWS Management Console 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. Puoi usare il filtro della tabella per visualizzare solo le KMS chiavi con tag particolari.

   Per visualizzare solo KMS le chiavi con un tag particolare, scegliete la casella del filtro, scegliete la chiave del tag, quindi scegliete uno dei valori effettivi del tag. È anche possibile digitare o tutto il valore del tag o solo parte di esso.

   La tabella risultante mostra tutte le KMS chiavi con il tag scelto. Tuttavia, il tag non viene visualizzato. Per visualizzare il tag, scegli l'ID o l'alias della KMS chiave 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 KMS le chiavi digitando solo la chiave del tag o solo il suo valore. Per filtrare i tag in base alla chiave o al valore del tag, usa l'ListResourceTagsoperazione per ottenere KMS le chiavi con tag, quindi usa le funzionalità di filtro del tuo linguaggio di programmazione.
5. Seleziona la casella di controllo accanto all'alias di una KMS chiave.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della KMS chiave, scegli la scheda Tag.

Usando il AWS KMS API

L'ListResourceTagsoperazione ottiene i tag per una KMS chiave. Il parametro `KeyId` è obbligatorio. Non è possibile utilizzare questa operazione per visualizzare i tag sui KMS tasti in un modo diverso Account AWS.

Ad esempio, il comando seguente ottiene i tag per una KMS chiave 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"
    }
  ]
}
```

Utilizzate i tag per controllare l'accesso alle KMS chiavi

Puoi controllare l'accesso a in AWS KMS keys base ai tag sulla KMS chiave. Ad esempio, puoi scrivere una IAM policy che consenta ai principali di abilitare e disabilitare solo le KMS chiavi che hanno un tag particolare. Oppure è possibile utilizzare una IAM policy per impedire ai principali di utilizzare KMS le chiavi nelle operazioni crittografiche a meno che la KMS chiave non abbia un tag particolare.

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, vedi A cosa serve? ABAC AWS e Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse nella Guida IAM per l'utente. Per assistenza nella risoluzione dei problemi di accesso relativi aABAC, vedereRisoluzione dei problemi per ABAC AWS KMS.

> **Note**
>
> Potrebbero essere necessari fino a cinque minuti prima che le modifiche ai tag e agli alias influiscano sull'autorizzazione delle KMS chiavi. Le modifiche recenti potrebbero essere visibili nelle API operazioni prima che influiscano sull'autorizzazione.

AWS KMS supporta la chiave di contesto della condizione globale aws:ResourceTag/tag-key, che consente di controllare l'accesso alle KMS chiavi in base ai tag sulla KMS chiave. Poiché più KMS chiavi possono avere lo stesso tag, questa funzionalità consente di applicare l'autorizzazione a un set selezionato di KMS chiavi. Puoi anche cambiare facilmente le KMS chiavi del set cambiando i relativi tag.

In AWS KMS, la chiave di `aws:ResourceTag/`*`tag-key`* condizione è supportata solo nelle IAM politiche. Non è supportata nelle politiche chiave, che si applicano solo a una KMS chiave, o nelle operazioni che non utilizzano una KMS chiave particolare, come le ListAliasesoperazioni ListKeyso.

Il controllo dell'accesso con i tag offre un modo semplice, scalabile e flessibile per gestire le autorizzazioni. Tuttavia, se non è progettato e gestito correttamente, può consentire o negare l'accesso alle KMS chiavi 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. Concedi IAM ai responsabili solo le autorizzazioni di cui hanno bisogno solo sulle KMS chiavi che devono usare o gestire. Ad esempio, usa i tag per etichettare le KMS chiavi utilizzate per un progetto. Quindi concedi al team di progetto il permesso di utilizzare solo KMS le chiavi con il tag del progetto.
- Fai attenzione a dare ai principali le autorizzazioni `kms:TagResource` e `kms:UntagResource` che consentono di aggiungere, modificare ed eliminare tag. Quando usi i tag per controllare l'accesso alle KMS chiavi, la modifica di un tag può dare ai responsabili il permesso di usare KMS chiavi che altrimenti non avrebbero il permesso di usare. Può anche negare l'accesso alle KMS chiavi di cui altri dirigenti hanno bisogno per svolgere il proprio lavoro. Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le politiche chiave o creare sovvenzioni possono controllare l'accesso alle KMS chiavi se dispongono dell'autorizzazione per gestire i tag.

  Quando possibile, utilizza una condizione politica, ad esempio `aws:RequestTag/`*`tag-key`* o `aws:TagKeys` per limitare le autorizzazioni di etichettatura del principale a tag o modelli di tag particolari su chiavi particolari. KMS
- Rivedi i principi del tuo sistema Account AWS che attualmente dispongono delle autorizzazioni di etichettatura e rimozione dei tag e modificali, se necessario. Ad esempio, la politica delle chiavi predefinita della console per gli amministratori chiave include `kms:TagResource` l'autorizzazione relativa a tale chiave. `kms:UntagResource` KMS IAMle politiche potrebbero consentire le autorizzazioni di etichettatura e rimozione dei tag su tutte le chiavi. KMS Ad esempio, la policy AWSKeyManagementServicePowerUsergestita consente ai principali di etichettare, rimuovere tag ed elencare i tag su tutte le chiavi. KMS
- Prima di impostare una politica che dipende da un tag, esamina i tag sulle KMS chiavi del tuo. Account AWS Assicurati che la tua policy si applichi solo ai tag che intendi includere. Usa CloudTrail registri e CloudWatch allarmi per avvisarti delle modifiche ai tag che potrebbero influire sull'accesso alle tue KMS chiavi.
- 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, considera la seguente IAM politica. Consente ai responsabili di richiamare le operazioni GenerateDataKeyWithoutPlaintexte decriptare solo sulle KMS chiavi 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.

```
{
  "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 IAM politica di esempio consente ai principali di utilizzare qualsiasi KMS chiave dell'account per determinate operazioni crittografiche. Ma proibisce ai principali di utilizzare queste operazioni crittografiche su KMS chiavi con o senza tag. `"Type"="Reserved"` `"Type"`

```
{
  "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 moduli crittografici convalidati NIST Federal Information Processing Standards (FIPS) 140 Cryptographic Module Validation Program (FIPS) 140-2 Level 3. Il materiale chiave per le tue chiavi non esce mai non crittografato. KMS HSMs

AWS KMS supporta diversi tipi di archivi di chiavi per proteggere le chiavi. Materiale da utilizzare AWS KMS per creare e gestire le chiavi di crittografia. Tutte le opzioni di archiviazione delle chiavi fornite da AWS KMS sono continuamente convalidate al di sotto del livello di sicurezza FIPS 140 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 la vostra autorizzazione.

AWS KMS archivio chiavi standard

Per impostazione predefinita, una KMS chiave viene creata utilizzando lo standard AWS KMS HSM. Questo HSM tipo può essere considerato come una flotta multi-tenant HSMs che offre l'archivio di chiavi più scalabile, più economico e più semplice da gestire dal proprio punto di vista. Se desideri creare una KMS chiave da utilizzare all'interno di una o più chiavi Servizi AWS in modo che il servizio possa crittografare i dati per tuo conto, creerai una chiave simmetrica. Se utilizzi una KMS chiave 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, potete scegliere di importare il materiale chiave in cui generare la vostra chiave di

crittografia simmetrica a 256 bit AWS KMS, o chiave ellittica () RSA o chiave Hash Based Message Authentication Code (ECC), e applicarla a un identificatore di chiave (HMAC). KMS keyId A volte questa operazione viene chiamata 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 solo una AWS KMS HSM volta che ha lasciato l'ambiente.

Il materiale relativo alle chiavi importato può essere utile se avete requisiti specifici relativi al sistema che genera le chiavi o se desiderate una copia della chiave non utilizzata AWS come backup. 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'opzione speciale API per l'eliminazione: DeleteImportedKeyMaterial. Ciò API eliminerà immediatamente tutte le copie del materiale chiave AWS KMS importato che contiene, senza possibilità AWS di recuperare la chiave. Inoltre, è possibile impostare una data di scadenza per una chiave importata, dopo la quale la chiave sarà inutilizzabile. Per rendere nuovamente utile la chiave AWS KMS, dovrai 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 verranno KMS effettivamente eliminati 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 AWS KMS sviluppatori.

Esistono alcune limitazioni aggiuntive relative al materiale chiave importato di cui tenere conto. Poiché AWS KMS non è possibile generare nuovo materiale per le chiavi, non è possibile configurare la rotazione automatica delle chiavi importate. È necessario creare una nuova KMS chiave con un nuovo materiale chiavekeyId, 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 prevista nell'ambito di una precedente operazione di crittografia. KMS 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 oECC) 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 diHSMs, è possibile creare un archivio di 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 si utilizza una KMS chiave in un archivio di chiavi personalizzato, le operazioni crittografiche vengono eseguite dal gestore delle chiavi utilizzando 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

Possedere un account HSMs può essere utile per soddisfare determinati requisiti normativi che non consentono ancora ai servizi web multi-tenant, come l'archivio di KMS chiavi standard, di conservare le chiavi crittografiche. Gli archivi di chiavi personalizzati non sono più sicuri degli archivi di KMS chiavi che utilizzano AWS-managedHSMs, 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.

- Tasti asimmetrici KMS
- HMACKMSchiavi

- KMSchiavi con materiale chiave importato
- Rotazione automatica delle chiavi
- Chiavi multi-regione

AWS CloudHSM negozio di chiavi

È possibile creare una KMS chiave in un AWS CloudHSMarchivio di chiavi, in cui le chiavi utente root vengono generate, archiviate e utilizzate in un AWS CloudHSM cluster di proprietà e gestione dell'utente. 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 KMS chiavi in un AWS CloudHSM cluster. Per vedere se un archivio chiavi AWS CloudHSM personalizzato è adatto alle tue esigenze, leggi Gli archivi 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 archivio di chiavi esterno (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 XKS proxy della rete, che quindi inoltra la richiesta al sistema crittografico utilizzato. Il XKS Proxy 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 KMS chiave in un AWS CloudHSM cluster di tua proprietà e gestione. Quando si utilizza una KMS chiave in un archivio di chiavi personalizzato, le operazioni crittografiche vengono eseguite nel HSMs cluster. Questa funzionalità

combina la praticità e l'ampia integrazione AWS KMS con il controllo aggiuntivo di un AWS CloudHSM cluster all'interno dell'azienda Account AWS.

AWS KMS offre console e API supporto completi per la creazione, l'utilizzo e la gestione degli archivi di chiavi personalizzati. È possibile utilizzare le KMS chiavi nell'archivio chiavi personalizzato nello stesso modo in cui si utilizza qualsiasi KMS chiave. Ad esempio, puoi utilizzare le KMS chiavi per generare chiavi di dati e crittografare i dati. Puoi anche utilizzare KMS le chiavi 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 FIPS140-2 moduli crittografici convalidati, soddisfa i requisiti di sicurezza. Non è richiesto un livello supplementare di responsabilità per la manutenzione o la dipendenza da un ulteriore servizio.

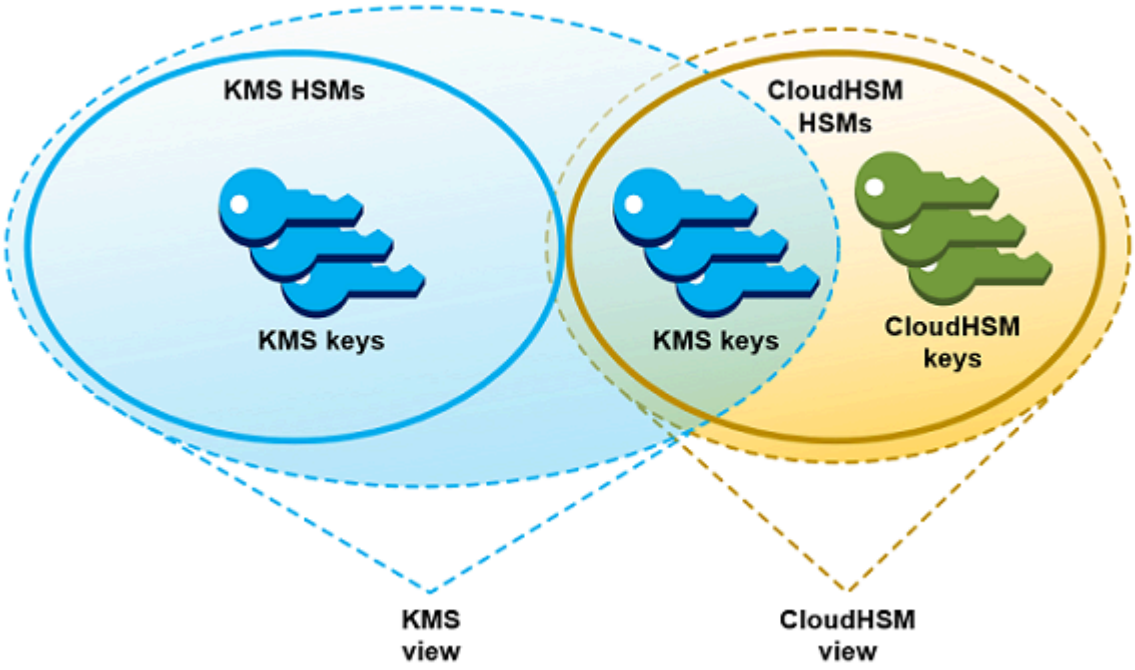
Tuttavia, è possibile prendere in considerazione la creazione di uno store di chiavi personalizzato qualora l'organizzazione possieda i seguenti requisiti:

- Hai delle chiavi che devono essere protette in modo esplicito in un singolo tenant HSM o in un HSM appartamento 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 HSM cluster in. AWS CloudHSM Quando crei AWS KMS keys in un archivio di chiavi AWS KMS personalizzato, visualizzi e gestisci le KMS chiavi in AWS KMS. Tuttavia, in AWS CloudHSMè anche possibile visualizzare e gestire il proprio materiale chiave, esattamente come avviene per le altre chiavi nel cluster.



È possibile creare KMS chiavi di crittografia simmetriche con materiale chiave generato dal proprio AWS KMS archivio di chiavi personalizzato. Utilizza quindi le stesse tecniche per visualizzare e gestire le KMS chiavi nell'archivio chiavi personalizzato utilizzate per KMS le chiavi nell'archivio AWS KMS chiavi. Puoi controllare l'accesso IAM e le politiche chiave, creare tag e alias, abilitare e disabilitare le KMS chiavi e pianificare l'eliminazione delle chiavi. È possibile utilizzare le KMS chiavi per operazioni crittografiche e utilizzarle con AWS servizi che si integrano con. AWS KMS

Inoltre, hai il pieno controllo sul AWS CloudHSM cluster, inclusa la creazione, l'eliminazione HSMs e la gestione dei backup. È possibile utilizzare il AWS CloudHSM client e le librerie software supportate per visualizzare, controllare e gestire il materiale chiave per le KMS chiavi. Sebbene l'archivio chiavi personalizzato sia disconnesso, AWS KMS non possono accedervi e gli utenti non possono utilizzare le KMS chiavi nell'archivio chiavi personalizzato per operazioni di crittografia. 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 AWS KMS e. 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 di crittografia simmetriche nel tuo archivio di KMS chiavi personalizzato. Basta specificare l'archivio chiavi personalizzato quando si crea la KMS chiave.

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 KMS chiavi in un archivio di AWS CloudHSM 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 eHSMs, consulta la sezione AWS CloudHSM Prezzi.

Regioni

AWS KMS supporta i AWS CloudHSM principali negozi 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.

- Tasti asimmetrici KMS
- HMACKMSchiavi
- KMSchiavi con materiale chiave importato
- Rotazione automatica delle chiavi
- Chiavi multi-regione

AWS CloudHSM concetti chiave del negozio

In questo argomento vengono descritti alcuni termini e concetti utilizzati nei AWS CloudHSM key store.

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 FIPS140-2 di livello 3.

Quando si crea una KMS chiave nel proprio 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 KMS chiave 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 KMS le chiavi AWS KMS mantenendo il pieno controllo del materiale su HSMs cui sono archiviate le chiavi, inclusa la gestione di cluster e backup. HSMs È possibile utilizzare la AWS KMS console e APIs gestire l'archivio delle AWS CloudHSM chiavi e le relative KMS chiavi. È inoltre possibile utilizzare la AWS CloudHSM consoleAPIs, 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 è necessario eliminare un archivio AWS CloudHSM chiavi, è necessario innanzitutto eliminare le KMS chiavi nell'archivio AWS CloudHSM chiavi pianificandone l'eliminazione e attendendo 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 si utilizza una KMS chiave nell'archivio delle AWS CloudHSM chiavi, 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 KMS chiavi nel AWS CloudHSM key store, il cluster associato deve contenerne almeno due attiveHSMs. Tutte le altre operazioni ne richiedono solo unaHSM.

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 eliminareHSMs, 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 è connesso un archivio chiavi personalizzato, è possibile creare e utilizzare KMS le relative chiavi. Quando è disconnesso, è possibile visualizzare e gestire l'archivio delle AWS CloudHSM chiavi e KMS le relative chiavi. Tuttavia, non è possibile creare nuove KMS chiavi o utilizzare le KMS chiavi 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 (CU) dedicato 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 in Cloud. HSM CLI 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 archiviarla 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`.

KMSchiavi in un archivio di AWS CloudHSM chiavi

È possibile utilizzare AWS KMS o AWS KMS API per creare un file AWS KMS keys in un archivio di AWS CloudHSM chiavi. Usi la stessa tecnica che useresti su qualsiasi KMS chiave. 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 create una KMS chiave in un AWS CloudHSM key store, AWS KMS crea la chiave in AWS KMS e genera una KMS chiave simmetrica 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 basata sul cluster. AWS CloudHSM AES Sebbene AWS CloudHSM supportino chiavi simmetriche e asimmetriche di diversi tipi, gli archivi di chiavi supportano solo chiavi di crittografia simmetriche. AWS CloudHSM AES

È possibile visualizzare le KMS chiavi 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'DescribeKeyoperazione per trovare l'ID dell'archivio AWS CloudHSM chiavi e l'ID AWS CloudHSM del cluster.

Le KMS chiavi in un AWS CloudHSM key store funzionano esattamente come tutte KMS le chiavi in esso contenute AWS KMS. Gli utenti autorizzati necessitano delle stesse autorizzazioni per utilizzare

e gestire le KMS chiavi. Utilizzate le stesse procedure e API operazioni della console per visualizzare e gestire le KMS chiavi in un archivio di AWS CloudHSM chiavi. Queste includono l'attivazione e la disabilitazione KMS delle chiavi, la creazione e l'utilizzo di tag e alias e l'impostazione e la modifica delle IAM politiche chiave. È possibile utilizzare le KMS chiavi in un archivio di AWS CloudHSM 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 in un archivio di KMS chiavi. AWS CloudHSM

Lo stesso processo viene utilizzato anche per pianificare l'eliminazione di una KMS chiave in un archivio di AWS CloudHSM chiavi. Dopo la scadenza del periodo di attesa, AWS KMS elimina la KMS chiave da. KMS Quindi fa del suo meglio per eliminare il materiale chiave relativo alla KMS chiave 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 archivio di AWS CloudHSM chiavi

Utilizzi IAM le policy per controllare l'accesso al tuo archivio di AWS CloudHSM chiavi e al tuo AWS CloudHSM cluster. È possibile utilizzare policy, IAM policy e concessioni chiave per controllare l'accesso a tali file AWS KMS keys nel proprio 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 AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato 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 AWS CloudHSM delle chiavi richiedono la seguente autorizzazione per utilizzare le operazioni dell'archivio AWS CloudHSM API 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 AmazonEC2. Potrebbero inoltre aver bisogno di creareHSMs, 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 di AWS CloudHSM chiavi richiedono le stesse autorizzazioni di coloro che creano e gestiscono qualsiasi KMS chiave in. AWS KMS La politica delle chiavi predefinita per una KMS chiave in un archivio AWS CloudHSM chiavi è identica alla politica di chiave predefinita per KMS le chiavi in ingresso. AWS KMS Il controllo di accesso basato sugli attributi (ABAC), che utilizza tag e alias per controllare l'accesso alle KMS chiavi, è efficace anche sulle chiavi negli archivi di KMS chiavi. AWS CloudHSM
- I responsabili che utilizzano KMS le chiavi dell'archivio delle chiavi per le operazioni crittografiche necessitano dell'autorizzazione per eseguire l'operazione di crittografia con la AWS CloudHSM chiave, ad esempio KMS:Decrypt. KMS È possibile fornire queste autorizzazioni in una politica chiave, una politica. IAM Tuttavia, non hanno bisogno di autorizzazioni aggiuntive per utilizzare una KMS chiave in un archivio di AWS CloudHSM chiavi.

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**
>
> 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

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 KMS chiave, come un ID e Amazon Resource Name (ARN) in AWS KMS. Quindi crea il materiale chiave nel HSMs cluster 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 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'DescribeClustersoperazione.

  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, KMS le chiavi 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 che lo ha crittografato. KMS
- 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 TCP traffico sulle 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 diHSMs, utilizzare la AWS CloudHSM console o l'DescribeClustersoperazione. 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 di AWS CloudHSM Client SDK 5, Cloud HSM CLI. Il Cloud HSM CLI sostituisce `key-handle` con `key-reference`.
> Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

1. Segui le procedure introduttive descritte nell'argomento Guida introduttiva a Cloud HSM Command Line Interface (CLI) della Guida per l'AWS CloudHSM utente.
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'CreateCustomKeyStoreoperazione.

Utilizzo della console AWS KMS

Quando si crea un archivio di AWS CloudHSM chiavi in AWS Management Console, 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 a AWS Management Console 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.

Usando il AWS KMS API

È possibile utilizzare l'CreateCustomKeyStoreoperazione 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'DescribeCustomKeyStoresoperazione.

Utilizzo della AWS KMS console

Quando si visualizzano gli archivi delle AWS CloudHSM chiavi in AWS Management Console, 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 KMS chiave 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 difettosa KMS.

Per visualizzare gli archivi di AWS CloudHSM chiavi in un determinato account e in una determinata regione, utilizzare la procedura seguente.

1. Accedi a AWS Management Console 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).

Usando il AWS KMS API

Per visualizzare i tuoi archivi AWS CloudHSM chiave, usa l'DescribeCustomKeyStoresoperazione. 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 `CustomKeyStoreId` (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` of `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 KMS chiave in un archivio di AWS CloudHSM chiavi connesso falliscono, ciò potrebbe indicare un problema con l'archivio AWS CloudHSM chiavi o il relativo AWS CloudHSM cluster. Per assistenza, consulta Come correggere una chiave difettosa KMS.

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 KMS le chiavi negli archivi AWS CloudHSM delle chiavi
- Registrazione delle AWS KMS API chiamate con AWS CloudTrail

Modifica le impostazioni del AWS CloudHSM key store

È possibile modificare le impostazioni di un archivio di AWS CloudHSM chiavi 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 (KMSchiavi) nell'archivio chiavi personalizzato e non è possibile utilizzare le KMS chiavi in esso contenute per operazioni di crittografia.
2. Modifica una o più impostazioni dell'archivio AWS CloudHSM chiavi.

   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'DescribeClustersoperazione. 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'UpdateCustomKeyStoreoperazione.

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 AWS Management Console 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 dell'archivio delle AWS CloudHSM chiavi 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, è possibile gestire l'archivio AWS CloudHSM chiavi e KMS le relative chiavi, ma non è possibile creare o utilizzare KMS chiavi nell'archivio AWS CloudHSM 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.
   - Digitare l'ID del AWS CloudHSM cluster correlato.
   - Digita la password corrente dell'utente `kmsuser` crittografico nel AWS CloudHSM cluster associato.
7. Seleziona 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 KMS chiavi nell'archivio AWS CloudHSM chiavi o utilizzare le KMS chiavi nell'archivio AWS CloudHSM chiavi nelle operazioni crittografiche.

Usando il AWS KMS API

Per modificare le proprietà di un archivio di AWS CloudHSM chiavi, utilizzare l'UpdateCustomKeyStoreoperazione. Puoi modificare più proprietà di un store delle chiavi personalizzate nello stesso comando. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta di HTTP 200 e un JSON oggetto senza proprietà. Per verificare che le modifiche siano effettive, utilizzate l'DescribeCustomKeyStoresoperazione.

Gli esempi in questa sezione utilizzano AWS Command Line Interface (AWS CLI), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Inizia a utilizzare DisconnectCustomKeyStoreper 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 UpdateCustomKeyStoreper modificare il nome descrittivo del AWS CloudHSM key store in`DevelopmentKeys`. Il comando utilizza il `CustomKeyStoreId` 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 `CustomKeyStoreId` 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 è`ExamplePassword`. Il comando utilizza il `CustomKeyStoreId` 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. È possibile lasciare l'archivio chiavi personalizzato nello stato disconnesso, ma è necessario connetterlo prima di poter creare nuove KMS chiavi o utilizzare KMS chiavi esistenti per operazioni crittografiche. 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 ciascuno ENI è`KMS managed ENI for cluster <cluster-ID>`.

Il completamento del processo di connessione può richiedere fino a 20 minuti.

Prima di collegare il AWS CloudHSM key store, verificate che soddisfi i requisiti.

- Il AWS CloudHSM cluster associato deve contenere almeno un cluster attivoHSM. Per trovare il numero di HSMs nel cluster, visualizza il cluster nella AWS CloudHSM console o usa l'DescribeClustersoperazione. 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 DescribeCustomKeyStoresrisposta. 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 l'archivio AWS CloudHSM delle chiavi è connesso, è possibile creare KMS chiavi al suo interno e utilizzare KMS le chiavi 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'ConnectCustomKeyStoreoperazione.

Utilizzo della console AWS KMS

Per connettere un AWS CloudHSM key store in AWS Management Console, 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 AWS Management Console 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 “Creare una KMS chiave in un archivio di AWS CloudHSM chiavi”.

Usando il AWS KMS API

Per connettere un archivio di AWS CloudHSM chiavi disconnesso, utilizzare l'ConnectCustomKeyStoreoperazione. Il AWS CloudHSM cluster associato deve contenere almeno un cluster attivo HSM e lo stato della connessione non può esserloFAILED.

Il completamento del processo di connessione può richiedere fino a 20 minuti. A meno che non fallisca rapidamente, l'operazione restituisce una risposta di HTTP 200 e un JSON oggetto senza proprietà. Questa risposta iniziale non indica tuttavia che la connessione è riuscita. Per determinare lo stato di connessione dell'archivio chiavi personalizzato, consultate la DescribeCustomKeyStoresrisposta.

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'DescribeCustomKeyStoresoperazione 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'DescribeCustomKeyStoresoperazione. 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 di 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.

Quando un archivio AWS CloudHSM chiavi è disconnesso, è possibile gestire l'archivio AWS CloudHSM chiavi e KMS le relative chiavi, ma non è possibile creare o utilizzare KMS chiavi nell'archivio chiavi. AWS CloudHSM Lo stato di connessione dell'archivio chiavi è `DISCONNECTED` e lo stato delle KMS chiavi nell'archivio chiavi personalizzato è`Unavailable`, a meno che non lo

sianoPendingDeletion. È possibile 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 si disconnette un archivio chiavi personalizzato, le KMS chiavi nell'archivio chiavi diventano immediatamente inutilizzabili (a seconda dell'eventuale coerenza). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono modificate finché la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

> **Note**
>
> Sebbene un archivio chiavi personalizzato sia disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare KMS chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Per stimare meglio l'effetto della disconnessione dell'archivio chiavi personalizzato, identifica le KMS chiavi nell'archivio chiavi personalizzato e determinane l'uso passato.

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 KMS chiavi del AWS CloudHSM cluster. Quando si disconnette l'archivio di chiavi personalizzato, si AWS KMS disconnette dall'account utente

kmsuser crittografico nel AWS CloudHSM client. Ciò consente di accedere al cluster come kmsuser CU e di controllare e gestire il materiale chiave relativo alla chiave. KMS

- Per disabilitare immediatamente tutte le KMS chiavi nell'archivio delle AWS CloudHSM chiavi. È possibile disabilitare e riattivare KMS le chiavi in un AWS CloudHSM archivio di chiavi utilizzando l'DisableKeyoperazione AWS Management Console o. Queste operazioni vengono completate rapidamente, ma agiscono su una KMS chiave alla volta. La disconnessione dell'archivio AWS CloudHSM chiavi modifica immediatamente lo stato della chiave di tutte le KMS chiavi nell'archivio AWS CloudHSM chiavi inUnavailable, 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 èFAILED), 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'DisconnectCustomKeyStoreoperazione.

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 AWS Management Console 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 usando il AWS KMS API

Per disconnettere un archivio di AWS CloudHSM chiavi connesso, utilizzare l'DisconnectCustomKeyStoreoperazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta di HTTP 200 e un JSON oggetto 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 delle AWS CloudHSM chiavi sia disconnesso, utilizzare l'DescribeCustomKeyStoresoperazione. 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 si elimina un archivio AWS CloudHSM chiavi, AWS KMS elimina tutti i metadati relativi all'archivio AWS CloudHSM chiaviKMS, incluse le informazioni sulla sua associazione a un AWS CloudHSM cluster. Questa operazione non influisce sul AWS CloudHSM cluster, sui HSMs relativi 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 mai bisogno di usare nessuna delle KMS chiavi nell'archivio delle chiavi per alcuna operazione crittografica. Quindi pianifica l'eliminazione di tutte le KMS chiavi dal key store. Per informazioni su come trovare le KMS chiavi in un archivio di AWS CloudHSM chiavi, consultaTrova le KMS chiavi in un archivio di AWS CloudHSM chiavi.
- Conferma che tutte KMS le chiavi siano state eliminate. Per visualizzare le KMS chiavi in un archivio di AWS CloudHSM chiavi, vederethe section called “Identifica KMS le chiavi negli archivi AWS CloudHSM delle chiavi”.
- Disconnettere l'archivio di 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 KMS chiavi nell'archivio delle AWS CloudHSM chiavi. È possibile 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'DeleteCustomKeyStoreoperazione.

Utilizzo della AWS KMS console

Per eliminare un AWS CloudHSM key store in AWS Management Console, iniziate selezionando il AWS CloudHSM key store dalla pagina Custom key store.

1. Accedi a AWS Management Console 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.

Utilizzando il AWS KMS API

Per eliminare un archivio di AWS CloudHSM chiavi, utilizzare l'DeleteCustomKeyStoreoperazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta di HTTP 200 e un JSON oggetto senza proprietà.

Per iniziare, verificate che l'archivio delle AWS CloudHSM chiavi non ne contenga AWS KMS keys. Non è possibile eliminare un archivio chiavi personalizzato che contiene KMS chiavi. Il primo comando di esempio utilizza ListKeyse DescribeKeycerca AWS KMS keys nell'archivio delle AWS CloudHSM chiavi con l'esempio *cks-1234567890abcdef0* ID dell'archivio chiavi personalizzato. In questo caso, il comando non restituisce alcuna KMS chiave. In caso affermativo, utilizzate l'ScheduleKeyDeletionoperazione per pianificare l'eliminazione di ciascuna KMS chiave.

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

Quindi, disconnetti l'archivio delle AWS CloudHSM chiavi. Questo comando di esempio utilizza l'DisconnectCustomKeyStoreoperazione 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'DeleteCustomKeyStoreoperazione 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 i key store 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 le KMS chiavi non disponibili
- Come correggere una chiave difettosa KMS
- 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 KMS chiave
- Come accedere come kmsuser

Come correggere le KMS chiavi non disponibili

Lo stato della chiave AWS KMS keys in un archivio di AWS CloudHSM chiavi è in genere`Enabled`. Come tutte le KMS chiavi, lo stato della chiave cambia quando si disabilitano le KMS chiavi in un archivio AWS CloudHSM chiavi o se ne pianifica l'eliminazione. Tuttavia, a differenza di altre KMS chiavi, le KMS chiavi in un archivio chiavi personalizzato possono avere anche uno stato chiave di`Unavailable`.

Lo stato della chiave di `Unavailable` indica che la KMS chiave si trova in un archivio chiavi personalizzato che è stato intenzionalmente disconnesso e che i tentativi di riconnessione, se presenti, hanno avuto esito negativo. Sebbene una KMS chiave non sia disponibile, è possibile visualizzarla e gestirlaKMS, ma non utilizzarla per operazioni crittografiche.

Per trovare lo stato della KMS chiave, nella pagina Customer managed keys, visualizza il campo Stato della KMS chiave. Oppure, utilizza l'DescribeKeyoperazione e visualizza l'`KeyState`elemento nella risposta. Per informazioni dettagliate, consultare Identifica e visualizza le chiavi.

Le KMS chiavi in un archivio chiavi personalizzato disconnesso avranno lo stato chiave `Unavailable` o`PendingDeletion`. KMSle chiavi di cui è pianificata l'eliminazione da un archivio chiavi personalizzato hanno uno stato `Pending Deletion` chiave, anche quando l'archivio chiavi personalizzato è disconnesso. Ciò ti consente di annullare l'eliminazione pianificata delle chiavi senza riconnettere lo store delle chiavi personalizzate.

Per correggere una KMS chiave non disponibile, ricollega l'archivio chiavi personalizzato. Dopo la riconnessione dell'archivio chiavi personalizzato, lo stato delle KMS chiavi nell'archivio chiavi personalizzato viene automaticamente ripristinato allo stato precedente, ad `Enabled` esempio o. `Disabled` KMSle chiavi in attesa di eliminazione rimangono nello `PendingDeletion` stato. Tuttavia, sebbene il problema persista, l'attivazione e la disabilitazione di una KMS chiave 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 difettosa KMS

I problemi relativi alla creazione e all'utilizzo KMS delle AWS CloudHSM chiavi negli archivi delle chiavi possono essere causati da un problema relativo all'archivio delle AWS CloudHSM chiavi, al AWS CloudHSM cluster associato, alla KMS chiave o al materiale delle chiavi.

Quando un archivio AWS CloudHSM chiavi viene disconnesso dal relativo AWS CloudHSM cluster, lo stato delle KMS chiavi nell'archivio chiavi personalizzato è`Unavailable`. Tutte le richieste di creazione di KMS chiavi in un archivio di AWS CloudHSM chiavi disconnesso restituiscono un'`CustomKeyStoreInvalidStateException`eccezione. 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 KMS chiave in un archivio 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. `Connected` Ciò può essere dovuto a una qualsiasi delle condizioni seguenti.

- Il materiale chiave della KMS chiave potrebbe essere stato eliminato dal AWS CloudHSM cluster associato. Per indagare, trovate l'ID chiave del materiale chiave per una KMS chiave e, se necessario, provate a recuperare il materiale chiave.
- Tutti HSMs sono stati eliminati dal AWS CloudHSM cluster associato all'archivio delle AWS CloudHSM chiavi. Per utilizzare una KMS chiave in un archivio di AWS CloudHSM chiavi in un'operazione crittografica, il relativo AWS CloudHSM cluster deve contenerne almeno una attivaHSM. Per verificare il numero e lo stato HSMs di un AWS CloudHSM cluster, utilizza la AWS CloudHSM console o l'DescribeClustersoperazione. Per aggiungere un HSM file al cluster, usa la AWS CloudHSM console o l'CreateHsmoperazione.
- 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, consulta Come recuperare il materiale chiave eliminato per una KMS chiave.
- Il AWS CloudHSM cluster associato all'archivio di chiavi personalizzato non aveva alcuna sessione PKCS #11 disponibile. 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

messaggio `KMSInternalException` di errore relativo alle sessioni PKCS #11, arretra e riprova la richiesta.

Come correggere un errore di connessione

Se si tenta di connettere un archivio AWS CloudHSM chiavi 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'DescribeCustomKeyStoresoperazione.

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 il AWS CloudHSM key store, se necessario, e riprova a connetterti.

Quando lo stato della connessione è `FAILED` impostato, esegui l'DescribeCustomKeyStoresoperazione 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 del cluster specificato. Ciò potrebbe verificarsi perché a un'APIoperazione è stato fornito un ID cluster errato o il cluster è stato eliminato e non sostituito. Per correggere questo errore, verifica l'ID del cluster, ad esempio utilizzando la AWS CloudHSM console o l'DescribeClustersoperazione. 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 connettersi, il cluster deve averne almeno unoHSM. Per trovare il numero di HSMs nel cluster, usa l'DescribeClustersoperazione. Per risolvere questo errore, aggiungine almeno uno HSM al cluster. Se ne aggiungi piùHSMs, 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 è possibile aggiungere indirizzi IP (CIDRblocchi) 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 sottoreti 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 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

- `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, disconnetti l'archivio delle AWS CloudHSM chiavi e usa il comando user change-password in Cloud HSM CLI 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'`kmsuser`account 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

Un'operazione crittografica che utilizza una KMS chiave in un archivio di chiavi personalizzato potrebbe non riuscire con un. `KMSInvalidStateException` L'eccezione `KMSInvalidStateException` può essere accompagnata dai seguenti messaggi di errore.

KMSnon può comunicare con il tuo HSM cluster Cloud. Potrebbe trattarsi di un problema di rete temporaneo. Se vedi ripetutamente questo errore, verifica che la rete ACLs e le regole del gruppo VPC di sicurezza per il tuo AWS CloudHSM cluster siano corrette.

- Sebbene si tratti di un errore HTTPS 400, potrebbe derivare da problemi di rete temporanei. 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 del gruppo di VPC sicurezza che blocca il traffico in uscita.

KMSnon è possibile comunicare con il AWS CloudHSM cluster perché l'utente kmsuser è bloccato. Se vedi questo errore ripetutamente, disconnetti il AWS CloudHSM key store 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 DescribeCustomKeyStoresrisposta mostra un `ConnectionState` of `FAILED` e un `ConnectionErrorCode` valore di`INVALID_CREDENTIALS`, come illustrato 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 `kmsuser` CU. 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"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "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 DescribeCustomKeyStoresrisposta 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"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "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'DescribeCustomKeyStoresoperazione e visualizza il valore dell'`ConnectionErrorCode`elemento nella risposta.

- Se `ConnectionErrorCode` è `INVALID_CREDENTIALS`, determinare la password corrente per l'account `kmsuser`. Se necessario, usa il comando user change-password in Cloud HSM CLI per impostare la password su un valore noto.
- Se il `ConnectionErrorCode` valore è`USER_LOCKED_OUT`, è necessario utilizzare il comando user change-password in Cloud HSM CLI 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 KMS chiave da un archivio di AWS CloudHSM chiavi, potrebbe essere necessario eliminare manualmente il materiale chiave corrispondente dal cluster associato. AWS CloudHSM

Quando si crea una KMS chiave in un AWS CloudHSM key store, AWS KMS crea i metadati KMS chiave AWS KMS e genera il materiale chiave nel cluster associato AWS CloudHSM . Quando si pianifica l'eliminazione di una KMS chiave in un archivio di AWS CloudHSM chiavi, dopo il periodo di attesa, AWS KMS elimina i metadati della KMS chiave. Quindi AWS KMS fa del suo meglio per eliminare il materiale chiave corrispondente dal AWS CloudHSM cluster. 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 la KMS chiaveARN, l'ID del AWS CloudHSM 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 di AWS CloudHSM Client SDK 5, Cloud HSM CLI. Il Cloud HSM CLI sostituisce `key-handle` con`key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client

SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

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**
   >
   > Mentre un archivio chiavi personalizzato è disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare KMS chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Usa il comando key delete in Cloud HSM CLI per eliminare la chiave dal HSMs cluster.

   Tutte le voci di CloudTrail registro per le operazioni crittografiche con una KMS chiave in un archivio di AWS CloudHSM chiavi includono un `additionalEventData` campo con `customKeyStoreId` e`backingKey`. Il valore restituito nel `backingKeyId` campo è l'`id`attributo HSM chiave Cloud. Ti consigliamo di filtrare l'operazione di eliminazione delle chiavi `id` per eliminare il materiale chiave orfano che hai identificato nei tuoi CloudTrail log.

   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. **`0x`**`1a2b3c45678abcdef`

   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 inCome scollegarsi e riconnettersi.

Come recuperare il materiale chiave eliminato per una KMS chiave

Se il materiale chiave di una AWS KMS key viene eliminato, la KMS chiave è inutilizzabile e tutto il testo cifrato crittografato con la KMS chiave non può essere decrittografato. Ciò può accadere se il materiale chiave di una KMS chiave in un archivio di AWS CloudHSM chiavi viene eliminato dal cluster associato. AWS CloudHSM Potrebbe tuttavia essere possibile recuperare questo materiale.

Quando si crea una AWS KMS key (KMSchiave) in un archivio AWS CloudHSM chiavi, AWS KMS accede al AWS CloudHSM cluster associato e crea il materiale chiave per la KMS chiave. 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 KMS chiave viene eliminato da un cluster, lo stato della KMS chiave alla fine cambia HSMs in. `UNAVAILABLE` Se si tenta di utilizzare la KMS chiave per un'operazione di crittografia, l'operazione ha esito negativo con un'`KMSInvalidStateException`eccezione. Soprattutto, i dati crittografati con la KMS chiave non possono essere decrittografati.

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.

   Utilizzate l'DescribeBackupsoperazione 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 `kmsuser`Crypto User (CU). Crei 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 la Cloud HSM Command Line Interface () CLI.

> **Note**
>
> Sebbene un archivio chiavi personalizzato sia disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare KMS chiavi esistenti nelle operazioni crittografiche falliranno. 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 comekmsuser, 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 di AWS CloudHSM Client SDK 5, Cloud HSM CLI. Il Cloud HSM CLI sostituisce key-handle conkey-reference. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso. È possibile utilizzare la AWS KMS console o. AWS KMS API

   Mentre la tua AWS CloudHSM chiave è connessa, AWS KMS è connesso come. kmsuser Ciò impedisce l'accesso come kmsuser o la modifica della password kmsuser.

   Ad esempio, questo comando utilizza DisconnectCustomKeyStoreper 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 HSM CLI sezione Uso del cloud della Guida per l'AWS CloudHSM utente.

```
aws-cloudhsm > login --username admin --role admin
          Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. Usa il comando user change-password in Cloud HSM CLI per cambiare la password dell'`kmsuser`account con una che conosci. (AWS KMS ruota la password quando colleghi il tuo AWS CloudHSM key store.) 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 HSM CLI sezione Utilizzo del cloud della Guida AWS CloudHSM per l'utente.

```
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 di AWS CloudHSM Client SDK 5, Cloud HSM CLI. Il Cloud HSM CLI sostituisce `key-handle` con`key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando del Client SDK 3, la Cloud HSM 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 dal Client SDK 3 CMU e KMU al Client SDK 5 Cloud HSM CLI nella Guida per l'AWS CloudHSM utente.

1. Esegui l'operazione, quindi utilizza il comando logout in Cloud HSM CLI 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 ometti questo passaggio, non AWS KMS sarà possibile `kmsuser` accedere al cluster e tutti i tentativi di riconnessione dell'archivio chiavi personalizzato falliranno. È possibile utilizzare la AWS KMS console o il `KeyStorePassword` parametro dell'UpdateCustomKeyStoreoperazione.

   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'ConnectCustomKeyStoreoperazione 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'DescribeCustomKeyStoresoperazione per verificare che l'archivio 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": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "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 AWS le 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 crittografarli 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 KMS chiave 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 di chiavi esterno e non può creare, visualizzare, gestire o eliminare le chiavi. AWS KMS Interagisce invece solo con il software proxy (XKSproxy) di archiviazione delle chiavi esterno 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.

È possibile utilizzare KMS le chiavi in un archivio di chiavi esterno per la crittografia lato client, incluso con. AWS Encryption SDK Tuttavia, gli archivi di chiavi esterni sono una risorsa importante per la crittografia lato server, poiché consentono di proteggere più AWS risorse utilizzando chiavi Servizi AWS crittografiche esterne. AWS Servizi AWS che supportano le chiavi gestite dal cliente per la crittografia simmetrica supportano anche KMS le chiavi 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 KMS le chiavi dell'archivio chiavi esterno possono essere decrittografati solo utilizzando il gestore di chiavi esterno che controlli. Se revochi temporaneamente l'accesso al tuo gestore di chiavi esterno, ad esempio scollegando l'archivio chiavi esterno o disconnettendo il gestore di chiavi esterno dal proxy dell'archivio chiavi esterno, AWS perde ogni accesso alle tue chiavi crittografiche finché non le ripristini. Durante tale intervallo, il testo cifrato crittografato con le tue chiavi non può essere decrittografato. KMS Se revochi definitivamente l'accesso al tuo gestore di chiavi esterno, tutto il testo cifrato crittografato con una KMS chiave nell'archivio delle chiavi esterno diventa irrecuperabile. Le uniche eccezioni sono i AWS servizi che memorizzano brevemente nella cache le chiavi dati protette dalle chiavi dell'utente. KMS Queste chiavi dati continuano a funzionare fino alla disattivazione della risorsa o alla scadenza della cache. Per informazioni dettagliate, consultare In che modo le chiavi inutilizzabili influiscono sulle chiavi dati KMS.

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 FIPS140-2 moduli di sicurezza hardware convalidati con livello di sicurezza 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 KMS le chiavi con materiale chiave importato generato internamente HSMs e che puoi eliminare dalle KMS chiavi 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:

- È necessario utilizzare le chiavi crittografiche nel gestore delle chiavi locale o in un gestore di chiavi al di fuori del AWS proprio 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

KMSLe chiavi standard utilizzano materiale chiave generato e utilizzato in HSMs un ambiente di AWS KMS proprietà e gestione. Tu stabilisci le politiche di controllo degli accessi sulle tue KMS chiavi e configuri Servizi AWS che utilizzano KMS le chiavi 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 chiaviKMS.

KMSle chiavi negli archivi di chiavi esterni si basano sul materiale e sulle operazioni chiave del gestore delle chiavi esterno. 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 e del mantenimento dei nostri standard di sicurezza. Per garantire che ogni testo cifrato archiviato con chiavi esterne sia almeno altrettanto sicuro del testo AWS KMS cifrato standard, AWS KMS prima crittografa tutto il testo in chiaro con il materiale AWS KMS chiave specifico della KMS chiave, 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 di archiviazione chiavi AWS KMS esterno conforme alla API specifica del proxy dell'archivio chiavi esterno e della garanzia della disponibilità e della durabilità del materiale 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 i AWS 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 di chiavi esterni supportano solo chiavi di crittografia simmetriche. KMS All'interno AWS KMS, si utilizzano e gestiscono KMS le chiavi 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. Lo stesso viene utilizzato APIs con gli stessi parametri per richiedere un'operazione crittografica con una KMS chiave in un archivio di chiavi esterno che si utilizza per qualsiasi KMS chiave. Inoltre, il prezzo è lo stesso delle KMS chiavi standard. Per i dettagli, consulta KMSchiavi in archivi di chiavi esterni la sezione AWS Key Management Service Prezzi.

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 AWS CloudHSM chiavi e archivi chiavi esterni, indipendentemente dallo stato della connessione. Inoltre, sono previste quote di AWS KMS richiesta per l'uso delle KMS chiavi in un archivio di chiavi esterno.

Se si sceglie la connettività VPC proxy per il proxy dell'archivio chiavi esterno, potrebbero esserci anche delle quote per i 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 della rete () RTT di 35 millisecondi o meno.

Gli archivi di chiavi esterni sono supportati Regioni AWS in tutti i paesi ad eccezione della Cina (Pechino) e della Cina (Ningxia). AWS KMS

Caratteristiche non supportate

AWS KMS non supporta le seguenti funzionalità negli archivi di chiavi personalizzati.

- Tasti asimmetrici KMS
- HMACKMSchiavi

- KMSchiavi con materiale 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

Impara 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 KMS chiave in un archivio di chiavi esterno è associata a una chiave esterna nel gestore delle chiavi esterno. Quando si utilizza una KMS chiave in un archivio di chiavi esterno per la crittografia o la decrittografia, l'operazione viene eseguita nel gestore delle chiavi esterno utilizzando la chiave esterna, una disposizione nota come Hold your Own Keys ()HYOK. 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 di 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 dal AWS KMS cliente possono utilizzare le KMS chiavi dell'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 KMS chiavi in un archivio di chiavi esterno hanno modelli di fiducia, accordi di responsabilità condivisa e aspettative di prestazioni fondamentalmente diversi rispetto alle KMS chiavi 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 KMS chiavi in un archivio di chiavi esterno sono influenzate dall'hardware, dal software, dai componenti di rete e dalla distanza

tra il gestore delle chiavi esterno AWS KMS e il gestore delle chiavi. 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 che proteggi, puoi decidere quali richiedono una KMS chiave in un archivio di chiavi esterno e quali possono essere protette da una chiave standard. KMS Questo ti offre la flessibilità di scegliere KMS le chiavi per classificazioni di dati, applicazioni o progetti specifici.

Gestore delle chiavi esterne

Un gestore di chiavi esterno è un componente esterno in grado di AWS generare chiavi simmetriche a 256 bit ed eseguire la crittografia e la decrittografia AES simmetriche. Il gestore di chiavi esterno per un archivio chiavi esterno può essere un modulo di sicurezza hardware fisico (HSM), un gestore di chiavi virtuale HSM o 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 di chiavi esterno può essere supportato da un singolo gestore di chiavi esterno o da più istanze di gestore di chiavi correlate che condividono chiavi crittografiche, ad esempio un HSM cluster. 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 di chiavi esterno

Chiave esterna

Ogni KMS chiave in un archivio di chiavi esterno è associata a una chiave crittografica nel gestore di chiavi esterno nota come chiave esterna. Quando si esegue la crittografia o la decrittografia con una KMS chiave nell'archivio di chiavi esterno, l'operazione di crittografia viene eseguita nel gestore delle chiavi esterno utilizzando 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 cifrato crittografato con la KMS chiave associata non è recuperabile.

Per gli archivi di chiavi esterni, una chiave esterna deve essere una chiave a 256 bit abilitata e in grado di eseguire AES la crittografia e la decrittografia. Per maggiori dettagli sui requisiti della chiave esterna, consulta Requisiti per una KMS chiave in un archivio di chiavi esterno.

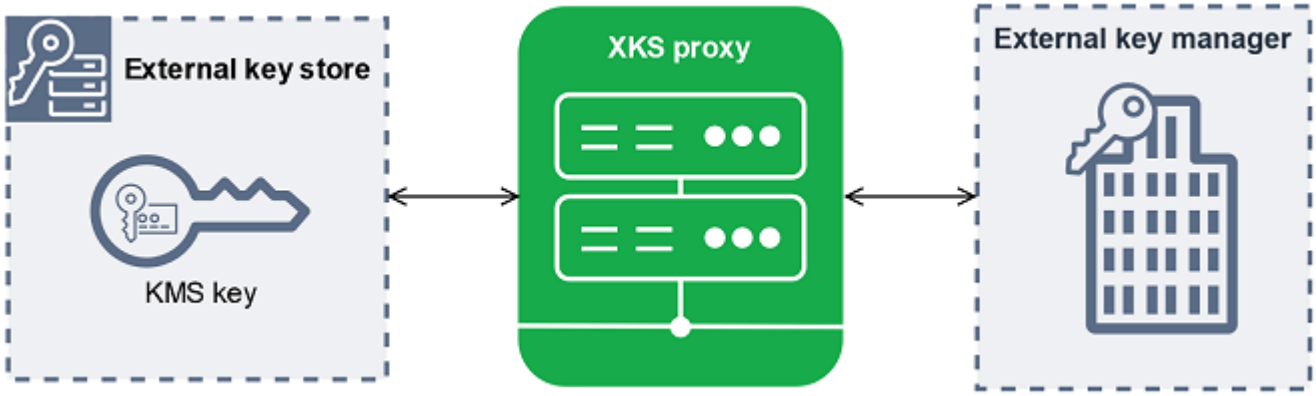
AWS KMS non può creare, eliminare o gestire chiavi esterne. Il materiale della tua chiave crittografica non esce mai dal tuo gestore di chiavi esterno. Quando crei una KMS chiave in un archivio di chiavi esterno, fornisci l'ID di una chiave esterna (). `XksKeyId` Non è possibile modificare l'ID della chiave esterna associata a una KMS chiave, sebbene il gestore delle chiavi esterno possa ruotare il materiale della chiave associato all'ID della chiave esterna.

Oltre alla chiave esterna, una KMS chiave in un archivio di chiavi esterno contiene anche materiale AWS KMS chiave. I dati protetti dalla KMS chiave vengono crittografati prima AWS KMS utilizzando il materiale della AWS KMS chiave, quindi dal gestore delle chiavi esterno utilizzando la chiave esterna. Questo processo di doppia crittografia garantisce che il testo cifrato protetto dalla KMS chiave sia sempre almeno altrettanto potente del testo cifrato protetto solo da. AWS KMS

Molte chiavi crittografiche presentano diversi tipi di identificatori. Quando crei una KMS chiave in un archivio di chiavi esterno, fornisci l'ID della chiave esterna che il proxy dell'archivio chiavi esterno utilizza per fare riferimento alla chiave esterna. Se utilizzi l'identificatore sbagliato, il tentativo di creare una KMS chiave nell'archivio di chiavi esterno fallisce.

Proxy dell'archivio delle chiavi esterne

Il proxy dell'archivio chiavi esterno («XKSproxy») è un'applicazione software di proprietà e gestita dal cliente che media tutte le comunicazioni tra AWS KMS e il gestore delle chiavi esterno. 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 APIspecifica del proxy di archiviazione 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 di archiviazione delle chiavi esterno sia conforme alle specifiche.

Per l'autenticazione AWS KMS, il proxy utilizza certificati lato server. TLS 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 mutual TLS (mTLS) per garantire ulteriormente che accetti solo richieste da. AWS KMS

Il proxy dell'archivio 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 GCM _256_ _ (1.3) SHA384 TLS
- TLS_ CHACHA2 0_ POLY13 SHA256 05_ (1,3) TLS

> **Note**
>
> Non AWS GovCloud (US) Region supporta TLS _ CHACHA2 0_ POLY13 05_. SHA256

- TLS_ _ ECDHE _ RSA _ WITH AES GCM _256_ _ (1.2) SHA384 TLS
- TLS_ _ ECDHE _ ECDSA _ WITH AES GCM _256_ _ (1,2) SHA384 TLS

Per creare e utilizzare le KMS chiavi nell'archivio chiavi esterno, è necessario innanzitutto connettere l'archivio chiavi esterno al relativo proxy dell'archivio chiavi esterno. Puoi anche disconnettere l'archivio delle chiavi esterne dal relativo proxy su richiesta. In tal caso, tutte le KMS chiavi dell'archivio chiavi esterno non sono più disponibili e non possono essere utilizzate in alcuna operazione crittografica.

Connettività proxy dell'archivio delle chiavi esterne

La connettività proxy dell'archivio chiavi esterno («connettività XKS proxy») descrive il metodo AWS KMS utilizzato per comunicare con il proxy dell'archivio chiavi esterno.

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 di archiviazione delle 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.
- VPCconnettività del servizio endpoint: AWS KMS invia le richieste a un servizio endpoint Amazon Virtual Private Cloud (AmazonVPC) creato e gestito da te. Puoi ospitare il tuo proxy di archiviazione chiavi esterno all'interno di Amazon VPC oppure ospitare il proxy dell'archivio chiavi esterno all'esterno AWS e utilizzare Amazon VPC solo per la comunicazione.

Per informazioni dettagliate sulle opzioni di connettività proxy dell'archivio delle chiavi esterne, consulta Scegli un'opzione di connettività proxy per l'archivio di chiavi esterno.

Credenziali di autenticazione al proxy dell'archivio delle chiavi esterne

Per autenticarti sul tuo proxy di archiviazione chiavi esterno, AWS KMS firma tutte le richieste sul proxy dell'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 XKS proxy non è correlata alle credenziali SigV4 associate ai principali presenti nel tuo. AWS Identity and Access Management Account AWS Non riutilizzate alcuna credenziale SigV4 per il proxy di archiviazione delle chiavi esternoIAM.

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 archiviarla.

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 API Specification.

Per consentire all'utente di ruotare le credenziali senza interrompere le Servizi AWS KMS chiavi utilizzate nell'archivio chiavi esterno, si consiglia 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 ulteriormente che il proxy dell'archivio chiavi esterno risponda solo a AWS KMS, alcuni proxy di chiavi esterni supportano Mutual Transport Layer Security (m). TLS Per informazioni dettagliate, consultare m TLS authentication (opzionale).

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 APIspecifica del proxy per l'archivio di chiavi AWS KMS esterne. Queste API richieste 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 API chiamate nelle CloudWatch metriche Amazon per gli archivi di chiavi esterni. Per informazioni dettagliate, consultare 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 al proxy API e tutte le eccezioni AWS KMS operative relative al proxyAPI.

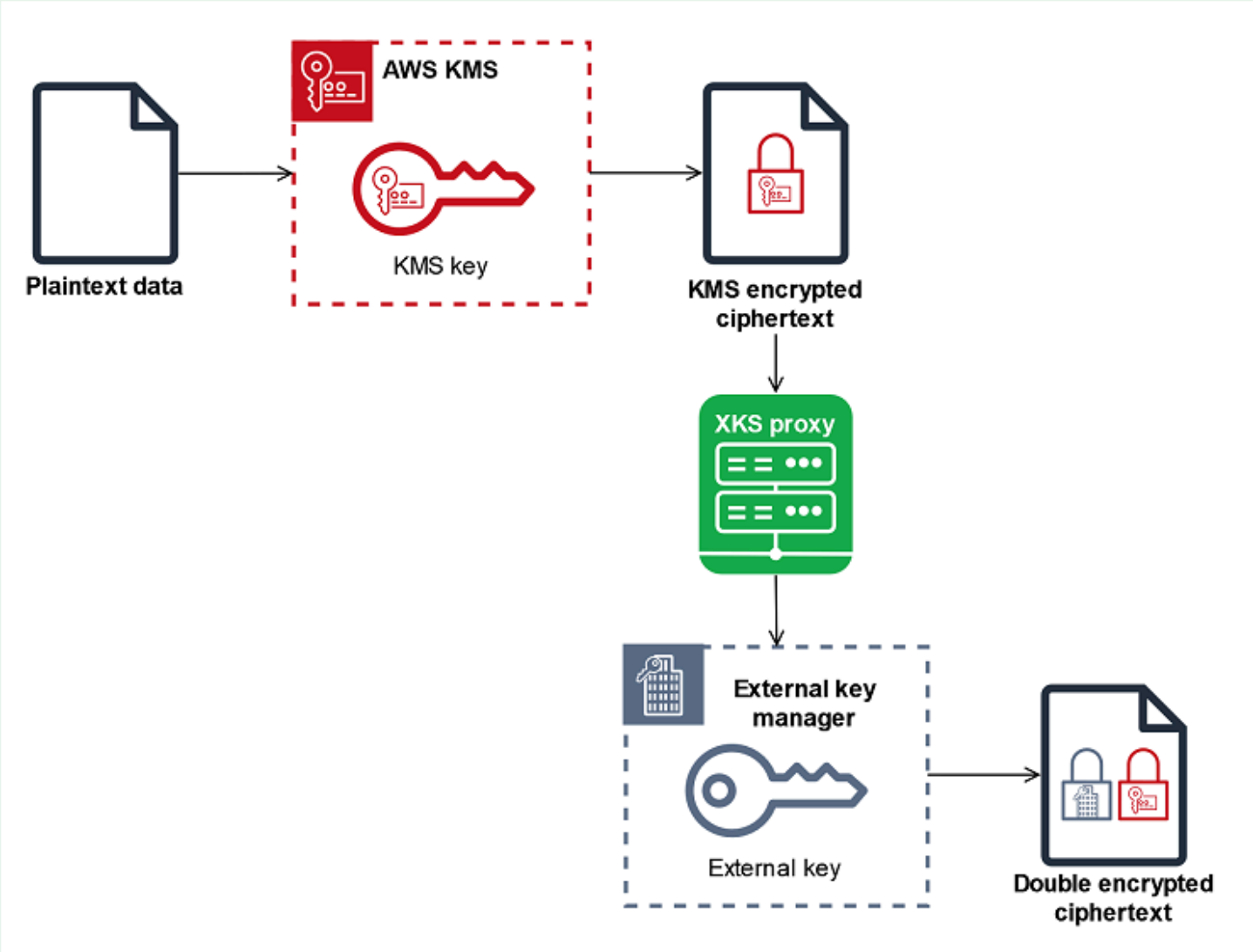
Proxy API	Descrizione	AWS KMS Operazioni correlate
Decrypt	AWS KMS invia il testo cifrato da decrittografare e l'ID della chiave esterna da utilizzare. L'algoritmo di crittografia richiesto è _. AES GCM	Decrittografa, ReEncrypt
Crittografa	AWS KMS invia i dati da crittografare e l'ID della chiave esterna da utilizzare. L'algoritmo di crittografia richiesto è AES _GCM.	Crittografa, GenerateDataKey, GenerateDataKeyWithoutPlaintextReEncrypt
GetHealthStatus	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 dei servizi VPC endpoint) Se tutte le istanze del gestore delle chiavi esterne sono `Unavailable` , i tentativi di creare o connettere l'archivio delle chiavi hanno esito negativo con l'eccezione `XksProxyUriUnreachableException` .
GetKeyMetadata	AWS KMS richiede informazioni sulla chiave esterna associata a una KMS chiave nell'archivio delle chiavi esterno. La risposta include le specifiche della chiave (`AES_256`), il suo utilizzo (`[ENCRYPT, DECRYPT]`) e	CreateKey Se la specifica della chiave non è `AES_256`, se l'utilizzo della chiave non è `[ENCRYPT, DECRYPT]` o lo stato è `DISABLED`, l'operazione `CreateKey` ha esito negativo con l'eccezione `XksKeyInvalidConfigurationException` .

Proxy API	Descrizione	AWS KMS Operazioni correlate
	se la chiave esterna è `ENABLED` o `DISABLED`.	

Doppia crittografia

I dati crittografati da una KMS chiave in un archivio di chiavi esterno vengono crittografati due volte. Innanzitutto, AWS KMS crittografa i dati con il materiale AWS KMS chiave specifico della KMS chiave. 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 KMS chiave in un archivio di chiavi esterno siano almeno altrettanto sicuri del testo cifrato crittografato da una chiave standard. KMS Protegge inoltre il testo non crittografato in transito dal proxy dell'archivio AWS KMS di 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 KMS chiave in un archivio di chiavi esterno dispone di due chiavi di supporto crittografiche:

- Un materiale AWS KMS chiave unico per la KMS chiave. Questo materiale chiave viene generato e utilizzato solo nei moduli di sicurezza hardware certificati AWS KMS FIPS140-2 Security Level 3 (HSMs).
- Una chiave esterna nel gestore delle chiavi esterne.

La doppia crittografia ha i seguenti effetti:

- AWS KMS non è possibile decrittografare alcun testo cifrato crittografato da una KMS chiave in un archivio di chiavi esterno senza accedere alle chiavi esterne tramite il proxy dell'archivio chiavi esterno.

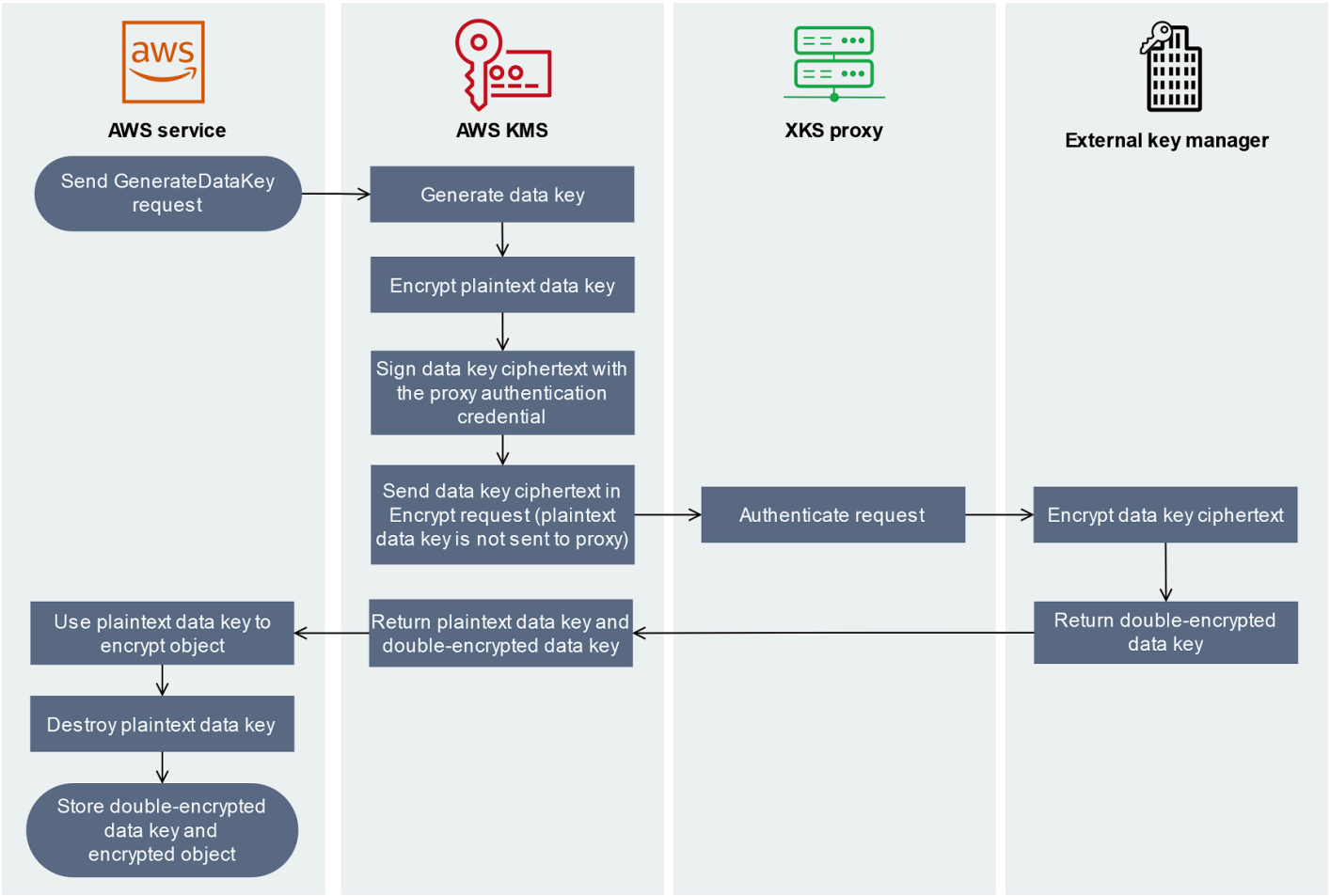
- Non è possibile decrittografare alcun testo cifrato crittografato da una KMS chiave in un archivio di chiavi esterno a AWS, anche se si dispone del relativo materiale di chiave esterno.
- Non è possibile ricreare una KMS chiave che è stata eliminata da un archivio di chiavi esterno, anche se si dispone del relativo materiale esterno. Ogni KMS chiave ha metadati unici che include nel testo cifrato simmetrico. Una nuova KMS chiave non sarebbe in grado di decrittografare il testo cifrato crittografato dalla chiave originale, anche se utilizzasse lo stesso materiale di 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 illustra il tipico flusso di lavoro di crittografia Servizio AWS che crittografa ogni oggetto con una chiave dati univoca protetta da una chiave. KMS In questo caso, hai scelto una KMS chiave in un archivio di chiavi esterno 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 KMS chiave in un archivio di chiavi esterno sia sempre almeno altrettanto efficace del testo cifrato crittografato da una chiave simmetrica standard contenente materiale KMS 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 GenerateDataKeyrichiesta all' AWS KMS utilizzo di una KMS chiave 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 testo semplice 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 nell'archivio di AWS KMS chiavi esterno. KMS
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 di 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 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, consultare In che modo le chiavi inutilizzabili influiscono sulle chiavi dati KMS.

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 (IAMpolitiche, politiche e concessioni chiave) utilizzate con KMS le chiavi standard funzionano allo stesso modo per KMS le chiavi in un archivio di chiavi esterno. È possibile utilizzare IAM le policy per controllare l'accesso alle API operazioni che creano e gestiscono archivi di chiavi esterni. IAMLe policy e le policy chiave vengono utilizzate per controllare l'accesso agli archivi di chiavi esterni. AWS KMS keys Puoi anche utilizzare le policy di controllo dei servizi per la tua AWS organizzazione e le policy VPC degli endpoint per controllare l'accesso alle KMS chiavi 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

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 principali archivi di altri. Account AWS

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms:DeleteCustomKeyStore`

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 di chiavi esterno con connettività ai servizi VPC endpoint, i responsabili devono disporre dell'autorizzazione a creare i seguenti componenti:

- Un Amazon VPC
- Sottoreti pubbliche e private
- Un Network Load Balancer e un gruppo di destinazione
- Un servizio di VPC endpoint Amazon

Per i dettagli, consulta Gestione delle identità e degli accessi per AmazonVPC, Gestione delle identità e degli accessi per VPC endpoint e servizi VPC endpoint e Autorizzazioni Elastic Load API Balancing.

Autorizzazione degli utenti delle chiavi in archivi di chiavi esterni KMS

I responsabili della creazione e della gestione AWS KMS keys nell'archivio di chiavi esterno richiedono le stesse autorizzazioni di coloro che creano e gestiscono qualsiasi KMS chiave in. AWS KMS La politica di chiave predefinita per la KMS chiave in un archivio di chiavi esterno è identica alla politica di chiave predefinita per KMS le chiavi in ingresso. AWS KMS Il controllo di accesso basato sugli attributi (ABAC), che utilizza tag e alias per controllare l'accesso alle KMS chiavi, è efficace anche sulle chiavi presenti in un archivio di KMS chiavi esterno.

I responsabili che utilizzano KMS le chiavi dell'archivio chiavi personalizzato per le operazioni crittografiche necessitano dell'autorizzazione per eseguire l'operazione di crittografia con la chiave, ad esempio KMS:Decrypt. KMS È possibile fornire queste autorizzazioni in una politica o chiave. IAM Tuttavia, non hanno bisogno di autorizzazioni aggiuntive per utilizzare una KMS chiave in un archivio di chiavi personalizzato.

Per impostare un'autorizzazione che si applica solo alle KMS chiavi in un archivio di chiavi esterno, utilizza la condizione della `kms:KeyOrigin`policy con un valore di`EXTERNAL_KEY_STORE`. È possibile utilizzare questa condizione per limitare l'CreateKeyautorizzazione kms: o qualsiasi autorizzazione specifica per una risorsa KMS chiave. Ad esempio, la seguente IAM politica consente all'identità a cui è associata di richiamare le operazioni specificate su tutte le KMS chiavi dell'account, a condizione che le KMS chiavi si trovino in un archivio di chiavi esterno. Si noti che è possibile limitare l'autorizzazione alle KMS chiavi in un archivio di chiavi esterno e alle KMS chiavi in un archivio di chiavi esterno dell'account Account AWS, ma non a nessun particolare archivio di chiavi esterno.

```
{
  "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 di archiviazione delle 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 le credenziali di autenticazione proxy dell'archivio chiavi esterno specificate 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 VPC endpoint, devi AWS KMS autorizzare la creazione di un endpoint di interfaccia per il tuo servizio VPC endpoint Amazon. Questa autorizzazione è richiesta indipendentemente dal fatto che il proxy dell'archivio chiavi esterno si trovi nel tuo account VPC o che il proxy dell'archivio chiavi esterno si trovi altrove, ma utilizzi il servizio VPC endpoint per comunicare. AWS KMS

AWS KMS Per consentire la creazione di un endpoint di interfaccia, utilizza la VPCconsole Amazon o l'ModifyVpcEndpointServicePermissionsoperazione. Consenti le autorizzazioni per il seguente principale: `cks.kms.<region>.amazonaws.com`.

Ad esempio, il AWS CLI comando seguente consente di connettersi AWS KMS al servizio VPC endpoint specificato nella regione Stati Uniti occidentali (Oregon) (us-west-2). Prima di utilizzare questo comando, sostituisci l'ID del VPC servizio Amazon 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 VPCconsole Amazon o il `RemoveAllowedPrincipals` parametro ModifyVpcEndpointServicePermissionswith the.

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 politiche, IAM dalle politiche e dalle concessioni chiave che autorizzano l'accesso alle operazioni che riguardano l'archivio di chiavi esterno o le relative chiavi. KMS

Per abilitare l'autorizzazione da parte del proxy dell'archivio chiavi esterno, AWS KMS include i metadati in ogni APIrichiesta proxy, tra cui il chiamante, la KMS chiave, l' AWS KMS operazione e Servizio AWS (se presente). I metadati della richiesta per la versione 1 (v1) del proxy API con chiave esterna 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.

m TLS authentication (opzionale)

Per consentire al proxy dell'archivio di 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 di 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. m TLS aggiunge l'autenticazione lato client, in cui il server proxy dell'archivio chiavi esterno autentica il AWS KMS client, all'autenticazione lato server TLS fornita dallo standard. Nel raro caso in cui le credenziali di autenticazione del proxy siano compromesse, m TLS impedisce a terzi di effettuare API richieste di successo al proxy dell'archivio chiavi esterno.

Per implementare mTLS, configura il tuo proxy di archiviazione delle chiavi esterno in modo che accetti solo TLS certificati lato client con le seguenti proprietà:

- Il nome comune dell'oggetto sul TLS certificato deve esserecks.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 di chiavi esterno

Prima di creare l'archivio di chiavi esterno, scegliete 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. 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 KMS chiavi supportate da materiale AWS KMS chiave.

> **Note**
>
> 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.

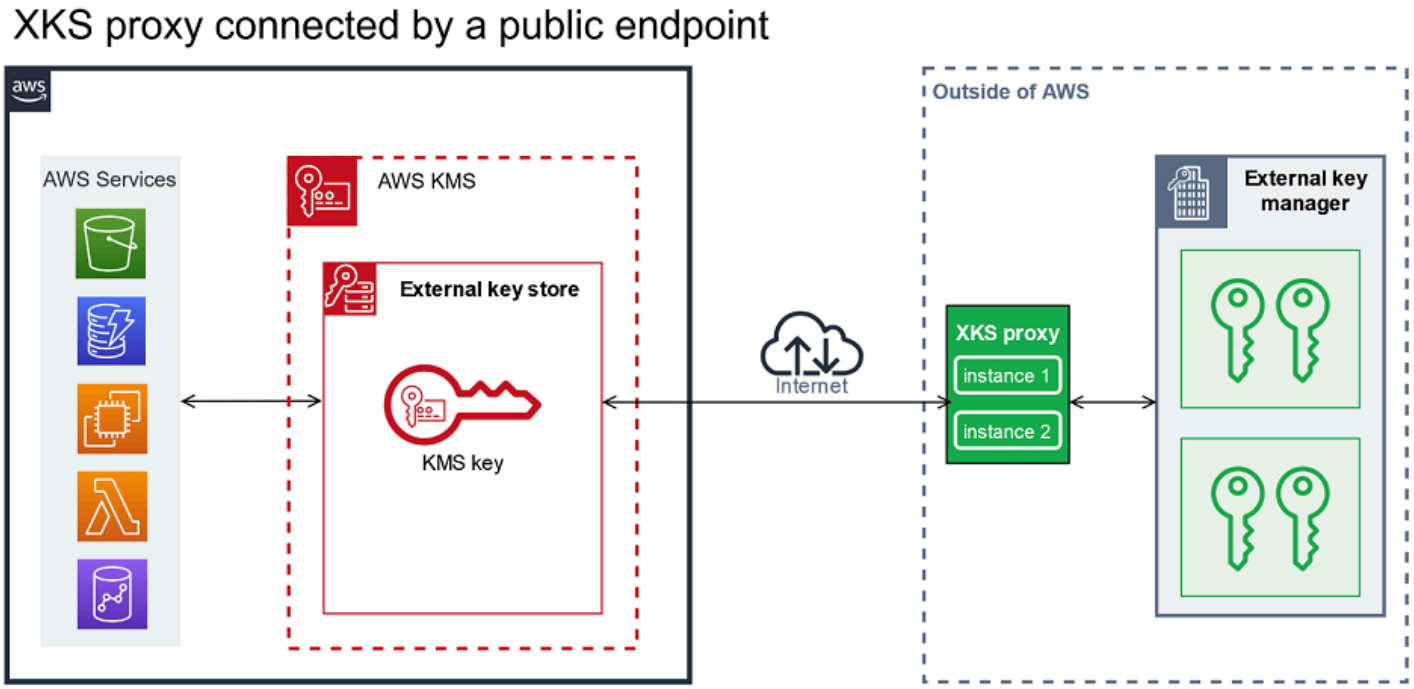
Puoi modificare l'opzione di connettività proxy dell'archivio delle chiavi esterne anche su un archivio delle chiavi esterne operativo. Tuttavia, il processo deve essere pianificato ed eseguito con cura per

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 (XKSproxy) 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.



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.
- È possibile utilizzare lo stesso endpoint pubblico per più archivi di chiavi esterni, a condizione che utilizzino valori di URIpercorso proxy diversi.
- 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 ai servizi VPC endpoint

nello stesso archivio Regione AWS, anche se gli archivi di chiavi si trovano in archivi diversi. Account AWS

- È necessario ottenere un TLS certificato emesso da un'autorità di certificazione pubblica supportata per gli archivi di chiavi esterni. Per un elenco, consulta Autorità di certificazione attendibili.

  Il nome comune dell'oggetto (CN) sul TLS certificato deve corrispondere al nome di dominio nell'URIendpoint proxy per il proxy dell'archivio chiavi esterno. Ad esempio, se l'endpoint pubblico è`https://myproxy.xks.example.com`, ilTLS, il CN sul TLS certificato deve essere `myproxy.xks.example.com` o. `*.xks.example.com`

- Assicurati che tutti i firewall tra AWS KMS e il proxy di archiviazione delle chiavi esterno consentano il traffico da e verso la porta 443 sul proxy. AWS KMS comunica sulla porta 443. Questo valore non è configurabile.

Per informazioni su tutti i requisiti di un archivio delle chiavi esterne, consulta Assemblare i prerequisiti.

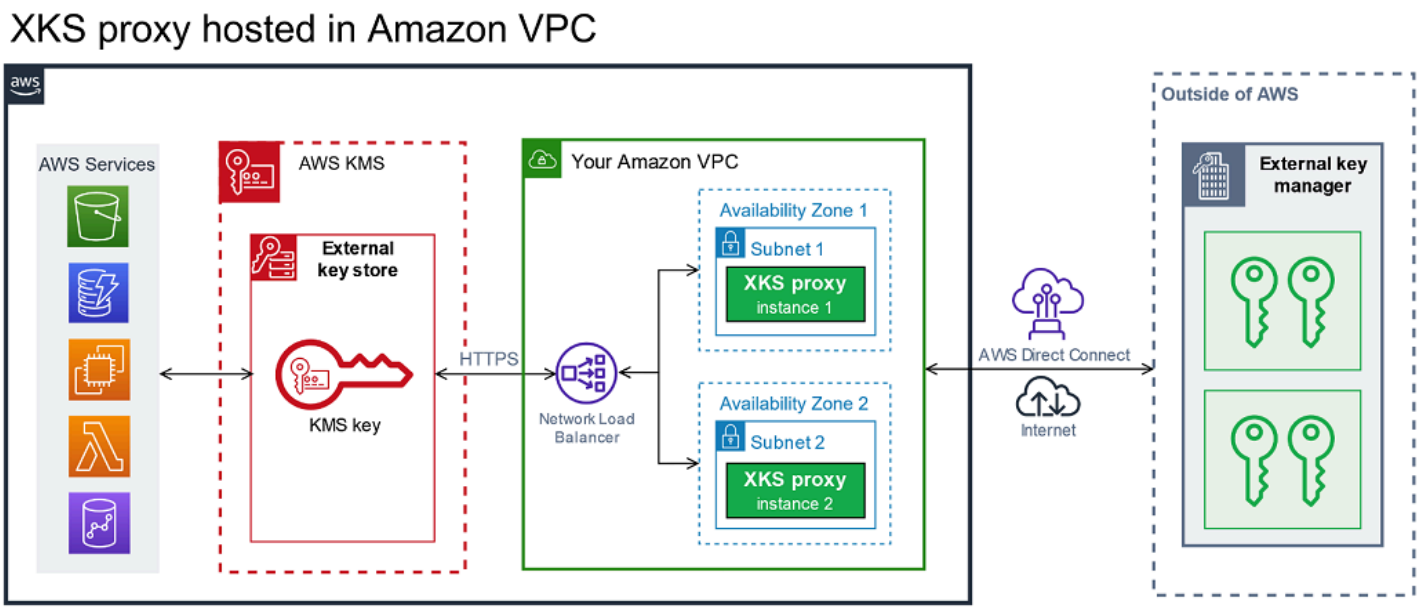
VPCconnettività dei servizi endpoint

AWS KMS si connette al proxy di archiviazione chiavi esterno (XKSproxy) creando un endpoint di interfaccia verso un servizio VPC endpoint Amazon creato e configurato da te. Sei responsabile della creazione del servizio VPC endpoint e della connessione VPC al tuo gestore di chiavi esterno.

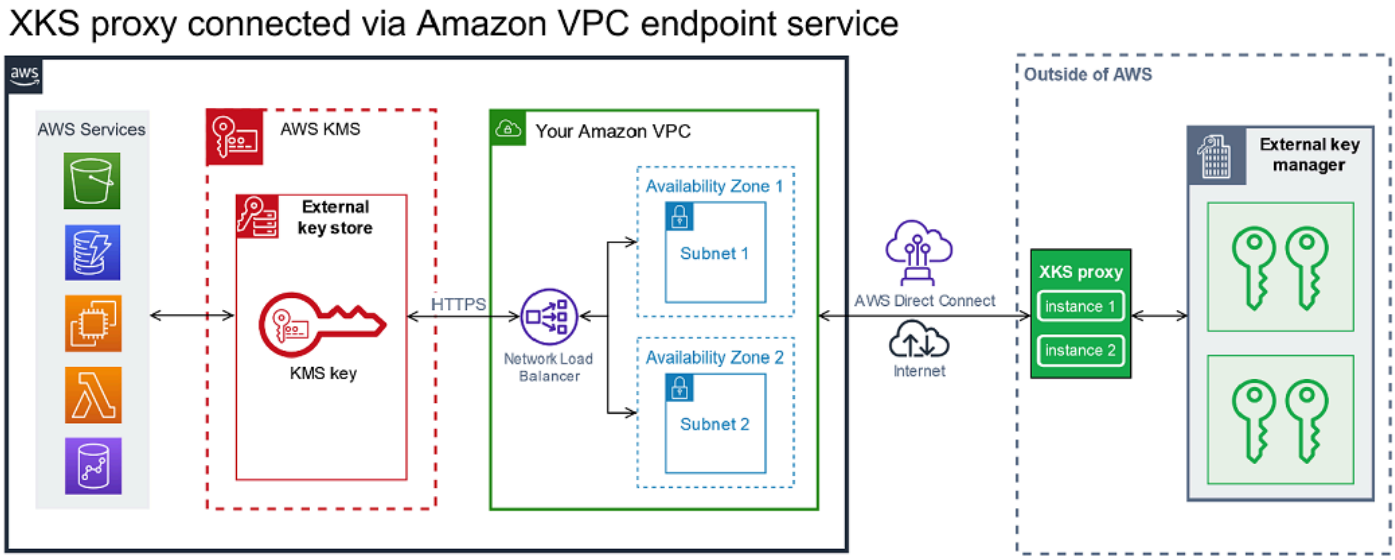
Il servizio endpoint può utilizzare tutte le network-to-AmazonVPCopzioni supportate per le comunicazioni, tra cui. AWS Direct Connect

Questa opzione di connettività è più complessa da configurare e gestire. Ma utilizza AWS PrivateLink, il che consente di AWS KMS connettersi privatamente ad Amazon VPC e al proxy di archiviazione chiavi esterno senza utilizzare la rete Internet pubblica.

Puoi localizzare il tuo proxy di archiviazione delle chiavi esterno in AmazonVPC.



Oppure, individua il tuo proxy di archiviazione delle chiavi esterno all'esterno AWS e utilizza il servizio di VPC endpoint Amazon solo per comunicazioni sicure con AWS KMS.



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.

Configura la VPC connettività del servizio endpoint

Utilizza le indicazioni contenute in questa sezione per creare e configurare le AWS risorse e i componenti correlati necessari per un archivio di chiavi esterno che utilizza la connettività del servizio VPC endpoint. 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 delle chiavi esterno in Amazon VPC o localizzare il proxy all'esterno AWS e utilizzare il servizio VPC endpoint per la comunicazione.

Prima di iniziare, verifica che sia necessario un archivio delle chiavi esterne. La maggior parte dei clienti può utilizzare KMS chiavi supportate da materiale AWS KMS chiave.

> **Note**
>
> Alcuni degli elementi necessari per la connettività del servizio VPC endpoint potrebbero essere inclusi nel gestore delle chiavi esterno. Inoltre, il software potrebbe avere requisiti di configurazione aggiuntivi. Prima di creare e configurare le AWS risorse in questa sezione, consultate la documentazione del proxy e del gestore delle chiavi.

Argomenti

- Requisiti per la VPC connettività dei servizi endpoint
- Passaggio 1: crea 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
- Passaggio 5: Verifica il tuo DNS nome di dominio privato
- Passaggio 6: AWS KMS Autorizza la connessione al servizio endpoint VPC

Requisiti per la VPC connettività dei servizi endpoint

Se si sceglie la connettività del servizio VPC endpoint per l'archivio di chiavi esterno, sono necessarie le seguenti risorse.

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 della rete (RTT) di 35 millisecondi o meno.

- Un Amazon VPC collegato al tuo gestore di chiavi esterno. Deve avere almeno due sottoreti private in due zone di disponibilità diverse.

  Puoi utilizzare un Amazon esistente VPC per il tuo archivio di chiavi esterno, a condizione che soddisfi i requisiti per l'utilizzo con un archivio di chiavi esterno. Più archivi di chiavi esterni possono condividere un AmazonVPC, ma ogni archivio di chiavi esterno deve avere il proprio servizio di VPC endpoint e il proprio DNS nome privato.
- Un servizio di VPC endpoint Amazon basato su un sistema AWS PrivateLink di bilanciamento del carico di rete e un gruppo target.

  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 DNS nome privato per il servizio VPC endpoint che è unico nel suo. Regione AWS

  Il DNS nome privato deve essere un sottodominio di un dominio pubblico di livello superiore. Ad esempio, se il DNS nome privato è`myproxy-private.xks.example.com`, deve essere un sottodominio di un dominio pubblico come o. `xks.example.com` `example.com`

  È necessario verificare la proprietà del DNS dominio per il DNS nome privato.
- Un TLS certificato rilasciato da un'autorità di certificazione pubblica supportata per il proxy di archiviazione delle chiavi esterno.

  Il nome comune dell'oggetto (CN) sul TLS certificato deve corrispondere al DNS nome privato. Ad esempio, se il DNS nome privato è`myproxy-private.xks.example.com`, il CN sul TLS certificato deve essere `myproxy-private.xks.example.com` o`*.xks.example.com`.

Per informazioni su tutti i requisiti di un archivio delle chiavi esterne, consulta Assemblare i prerequisiti.

Passaggio 1: crea un Amazon VPC e delle sottoreti

VPCla connettività dei servizi endpoint richiede un Amazon VPC connesso al tuo gestore di chiavi esterno con almeno due sottoreti private. Puoi creare un Amazon VPC o utilizzare un Amazon esistente VPC che soddisfi i requisiti per gli archivi di chiavi esterni. Per assistenza nella creazione di un nuovo AmazonVPC, consulta Create a VPC nella Amazon Virtual Private Cloud User Guide.

Requisiti per il tuo Amazon VPC

Per utilizzare archivi di chiavi esterni che utilizzano la connettività del servizio VPC endpoint, Amazon VPC deve avere le seguenti proprietà:

- Deve trovarsi nella stessa Account AWS regione supportata dell'archivio di chiavi esterno.
- Richiede almeno due sottoreti private, ognuna in una zona di disponibilità diversa.
- L'intervallo di indirizzi IP privati del tuo Amazon non VPC deve sovrapporsi all'intervallo di indirizzi IP privati del data center che ospita il tuo gestore di chiavi esterno.
- Tutti i componenti devono essere utilizzatiIPv4.

Hai molte opzioni per connettere Amazon VPC al tuo proxy di archiviazione chiavi esterno. Scegli un'opzione che soddisfi le tue esigenze di prestazioni e sicurezza. Per un elenco, vedi Connect your VPC ad altre reti e opzioni di Network-to-Amazon VPC connettività. Per ulteriori dettagli, consulta AWS Direct Connect e la Guida per l'utente di AWS Site-to-Site VPN.

Creare un Amazon VPC per il tuo archivio di chiavi esterno

Utilizza le seguenti istruzioni per creare Amazon VPC per il tuo archivio di chiavi esterno. Un Amazon VPC è necessario solo se scegli l'opzione di connettività del servizio VPC endpoint. Puoi utilizzare un Amazon esistente VPC che soddisfi i requisiti per un archivio di chiavi esterno.

Segui le istruzioni nell'argomento Creare un accountVPC, sottoreti e altre VPC risorse utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
IPv4CIDRb loccare	Inserisci gli indirizzi IP del tuoVPC. L'intervallo di indirizzi IP privati del tuo Amazon non VPC deve sovrapporsi all'intervallo di indirizzi IP privati del data center che ospita il tuo gestore di chiavi esterno.

Campo	Valore
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
NATgateway	Non è necessario indicare alcun valore.
VPCpunti finali	Non è necessario indicare alcun valore.
Abilita DNS i nomi host	Sì
Abilita la risoluzione DNS	Sì

Assicurati di testare la tua VPC comunicazione. Ad esempio, se il tuo proxy di archiviazione chiavi esterno non si trova in AmazonVPC, crea un'EC2istanza Amazon in AmazonVPC, verifica che Amazon sia in VPC grado di comunicare con il tuo proxy di archiviazione chiavi esterno.

Collegamento VPC di al gestore di chiavi esterno

Connettilo VPC al data center che ospita il tuo gestore di chiavi esterno utilizzando una qualsiasi delle opzioni di connettività di rete VPC supportate da Amazon. Assicurati che l'EC2istanza Amazon presente in VPC (o il proxy dell'archivio chiavi esterno, se presente inVPC) possa comunicare con il data center e il gestore delle chiavi esterno.

Fase 2: Creare un gruppo target

Prima di creare il servizio VPC endpoint richiesto, create i componenti richiesti, un sistema di bilanciamento del carico di rete (NLB) e un gruppo target. Il network load balancer (NLB) 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	Scegliete VPC dove creare il servizio VPC endpoint per il vostro archivio di chiavi esterno.
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 destinazione nella Guida per l'utente di Elastic Load Balancing per Network Load Balancer.
Rete	Altro indirizzo IP privato
IPv4indirizzo	Gli indirizzi privati del proxy dell'archivio delle chiavi esterne
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 VPC dove creare il servizio VPC endpoint per il tuo archivio di chiavi esterno.
Mapping	Scegli entrambe le zone di disponibilità (almeno due) configurate per le VPC sottoreti. 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 VPC endpoint, ne sei il fornitore e AWS KMS crei un endpoint per il tuo servizio. Per un archivio di chiavi esterno, crea un servizio VPC endpoint con il sistema di bilanciamento del carico di rete creato nel passaggio precedente. Il servizio VPC endpoint deve trovarsi nella stessa Account AWS regione supportata dell'archivio di chiavi esterno.

Più archivi di chiavi esterni possono condividere un AmazonVPC, ma ogni archivio di chiavi esterno deve avere il proprio servizio di VPC endpoint e il proprio DNS nome privato.

Segui le istruzioni nell'argomento Creare un servizio endpoint per creare il tuo servizio VPC endpoint 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 VPC endpoint senza un'accettazione manuale. Se è richiesta l'accettazione, i tentativi di creare l'archivio delle chiavi esterne falliscono con un'eccezione `XksProxyInvalidConfigurationException` .
Abilita il nome privato DNS	Associa un DNS nome privato al servizio
DNSNome privato	Inserisci un DNS nome privato che sia unico nel suo Regione AWS. Il DNS nome privato deve essere un sottodominio di un dominio pubblico di livello superiore. Ad esempio, se il DNS nome privato è`myproxy-private.xks.example.com` , deve essere un sottodominio di un dominio pubblico come `xks.example.com` o. `example.com` Questo DNS nome privato deve corrispondere al nome comune dell'oggetto (CN) nel TLS certificato configurato sul proxy di archiviazione delle chiavi esterno. Ad esempio, se il DNS nome privato è`myproxy-private.xks.example.com` , il CN sul TLS certificato deve essere `myproxy-private.xks.example.com` o `*.xks.example.com` .

Campo	Valore
	Se il certificato e il DNS nome privato non corrispondono, i tentativi di connetter e un archivio di chiavi esterno al relativo proxy di archiviazione chiavi esterno hanno esito negativo e viene visualizzato un codice di errore di connessio ne di`XKS_PROXY_INVALID_TLS_CONFIGURATION` . Per informazioni dettagliate, consultare Errori di configurazione generale.
Tipi di indirizzo IP supportati	IPv4

Passaggio 5: Verifica il tuo DNS nome di dominio privato

Quando crei il tuo servizio VPC endpoint, lo stato di verifica del dominio è`pendingVerification`. Prima di utilizzare il servizio VPC endpoint per creare un archivio di chiavi esterno, questo stato deve essere. `verified` Per verificare di essere il proprietario del dominio associato al proprio DNS nome privato, è necessario creare un TXT record in un DNS server pubblico.

Ad esempio, se il DNS nome privato del servizio VPC endpoint è`myproxy-private.xks.example.com`, devi creare un TXT record in un dominio pubblico, ad esempio `xks.example.com` o`example.com`, a seconda di quale dei due sia pubblico. AWS PrivateLink cerca il TXT record prima su `xks.example.com` e poi su. `example.com`

> **Tip**
>
> Dopo aver aggiunto un TXT record, potrebbero essere necessari alcuni minuti prima che il valore dello 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 VPCconsole Amazon, scegli Endpoint services e scegli il tuo endpoint service. Nel riquadro dei dettagli, vedi Domain verification status (Stato di verifica del dominio).
- Usa l'operazione. DescribeVpcEndpointServiceConfigurations Il valore `State` si trova nel campo `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Se lo stato di verifica non lo è`verified`, segui le istruzioni nell'argomento Verifica della proprietà del dominio per aggiungere un TXT record al DNS server del tuo dominio e verificare che il TXT record sia pubblicato. Quindi controlla nuovamente lo stato della verifica.

Non è necessario creare un record A per il nome di DNS dominio privato. Quando AWS KMS crea un endpoint di interfaccia verso il servizio VPC endpoint, crea AWS PrivateLink automaticamente una zona ospitata con il record A richiesto per il nome di dominio privato nel. AWS KMS VPC Per gli archivi di chiavi esterni con connettività al servizio VPC endpoint, ciò accade quando si collega l'archivio di chiavi esterno al relativo proxy di archiviazione chiavi esterno.

Passaggio 6: AWS KMS Autorizza la connessione al servizio endpoint VPC

È necessario aggiungerlo AWS KMS all'elenco dei principali indirizzi consentiti per il VPC servizio endpoint. Ciò consente di AWS KMS creare endpoint di interfaccia per il VPC servizio endpoint. Se non AWS KMS è un'opzione principale consentita, i tentativi di creare un archivio di chiavi esterno falliranno con un'`XksProxyVpcEndpointServiceNotFoundException`eccezione.

Segui le istruzioni nell'argomento Gestione delle autorizzazioni della Guida di AWS PrivateLink . Utilizza il seguente valore obbligatorio.

Campo	Valore
ARN	`cks.kms.<region>.amazonaws.com` Ad esempio, `cks.kms.us-east-1.amazonaws.com` .

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 (XKSproxy) che media la comunicazione tra AWS KMS e il gestore di chiavi esterno. Per informazioni dettagliate, consultare Scegli un'opzione di connettività proxy per l'archivio di chiavi esterno. Prima di iniziare, verifica che sia necessario un archivio delle chiavi esterne. La maggior parte dei clienti può utilizzare KMS chiavi supportate da materiale AWS KMS 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. Soprattutto, è necessario indicare se l'archivio chiavi esterno AWS KMS utilizza un endpoint pubblico o un servizio VPC endpoint per connettersi al proxy dell'archivio chiavi esterno. È inoltre necessario specificare i dettagli della connessione, tra cui l'URIendpoint del proxy e il percorso all'interno di tale endpoint proxy AWS KMS da cui invia API le richieste al proxy.

- Se utilizzi la connettività pubblica degli endpoint, assicurati che sia in AWS KMS grado di comunicare con il proxy tramite Internet utilizzando una connessione. HTTPS Ciò include la configurazione TLS sul proxy di archiviazione delle chiavi esterno e la garanzia che eventuali firewall tra AWS KMS e il proxy consentano il 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 VPC endpoint, assicurati che il servizio di bilanciamento del carico di rete, DNS il nome privato e il servizio VPC endpoint siano configurati correttamente e operativi. Se il proxy dell'archivio chiavi esterno non è presente inVPC, devi assicurarti che il servizio VPC endpoint possa comunicare con il proxy dell'archivio chiavi esterno. (AWS KMS verifica la connettività del servizio VPC endpoint quando si connette l'archivio chiavi esterno al relativo proxy di archiviazione chiavi esterno.)

Ulteriori considerazioni:

- 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 di utilizzare questi parametri per creare allarmi in grado di monitorare tale archivio, 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 KMS chiavi in un archivio di chiavi esterno è soggetto a quote di richiesta. 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 create il proxy di archiviazione delle chiavi esterno utilizzando AWS risorse, assicuratevi che il funzionamento del proxy non richieda la disponibilità di una KMS chiave in un archivio di chiavi esterno accessibile tramite tale proxy.

Tutti i nuovi archivi delle chiavi esterne vengono creati in uno stato disconnesso. Prima di poter creare KMS le chiavi nell'archivio di chiavi esterno, è necessario collegarlo al relativo proxy di archiviazione chiavi esterno. 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 di 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 JSON basato sul quale sono specificati il URIpercorso del proxy e 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 fisico o virtuale (HSM) 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 KMS le chiavi nell'archivio 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. 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.

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 KMS chiave 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 GetHealthStatusrisposta del proxy riporta che tutte le istanze del gestore di chiavi esterno

lo sonoUnavailable, tutti i tentativi di creare un archivio di chiavi esterno falliscono con un. XksProxyUriUnreachableException

Proxy dell'archivio delle chiavi esterne

È necessario specificare un proxy di archiviazione chiavi esterno (XKSproxy) conforme ai requisiti di progettazione indicati nella specifica del proxy API dell'archivio chiavi AWS KMS esterno. È possibile sviluppare o acquistare un proxy di archiviazione chiavi esterno oppure utilizzare un proxy di archiviazione chiavi esterno fornito o integrato nel gestore delle 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.

È possibile utilizzare un proxy di archiviazione chiavi esterno per più di un archivio chiavi esterno, ma ogni archivio di chiavi esterno deve avere un URI endpoint e un percorso univoci all'interno del proxy dell'archivio chiavi esterno per le sue richieste.

Se utilizzi la connettività del servizio VPC endpoint, puoi localizzare il tuo proxy di archiviazione chiavi esterno in AmazonVPC, ma non è necessario. Puoi localizzare il proxy all'esterno AWS, ad esempio nel tuo data center privato, e utilizzare il servizio VPC endpoint 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 archiviarla.

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 riutilizzate alcuna credenziale SigV4 per il proxy di archiviazione delle chiavi esternoIAM.

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 (AmazonVPC). 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 di VPC endpoint di Amazon, devi creare e gestire i componenti richiesti, tra cui un Amazon VPC con almeno due sottoreti in due zone di disponibilità diverse, un servizio VPC endpoint con un sistema di bilanciamento del carico di rete e un gruppo target e un DNS nome privato per il servizio endpoint. VPC

Puoi modificare l'opzione di connettività proxy dell'archivio delle chiavi esterne. Tuttavia, devi garantire la disponibilità continua del materiale chiave associato alle chiavi nel tuo archivio di KMS chiavi esterno. In caso contrario, AWS KMS non è possibile 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 di chiavi esterno. Per informazioni sulla creazione di una configurazione della connettività dei servizi VPC endpoint, consulta. Configura la VPC connettività del servizio endpoint

Endpoint proxy URI

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 essereHTTPS. AWS KMS comunica sulla porta 443. Non specificare la porta nel valore dell'URIendpoint del 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.
- VPCconnettività del servizio endpoint: specificare `https://` seguito dal DNS nome privato del servizio VPC endpoint.

Il certificato del TLS server configurato sul proxy dell'archivio chiavi esterno deve corrispondere al nome di dominio nell'URIendpoint proxy dell'archivio chiavi esterno ed essere rilasciato da un'autorità di certificazione supportata per gli archivi di chiavi esterni. Per un elenco, consulta Autorità di certificazione attendibili. L'autorità di certificazione richiederà una prova della proprietà del dominio prima di emettere il TLS certificato.

Il nome comune dell'oggetto (CN) sul TLS certificato deve corrispondere al DNS nome privato. Ad esempio, se il DNS nome privato è`myproxy-private.xks.example.com`, il CN sul TLS certificato deve essere `myproxy-private.xks.example.com` o `*.xks.example.com`.

Puoi modificare l'URIendpoint del proxy, ma assicurati che il proxy dell'archivio di chiavi esterno abbia accesso al materiale chiave associato alle KMS chiavi nell'archivio di chiavi esterno. In caso contrario, AWS KMS non è possibile decrittografare alcun testo cifrato crittografato con tali chiavi. KMS

Requisiti di unicità

- Il valore combinato del proxy URI endpoint (`XksProxyUriEndpoint`) e del URI percorso proxy (`XksProxyUriPath`) deve essere univoco nella regione and. Account AWS
- Gli archivi di chiavi esterni con connettività pubblica agli endpoint possono condividere lo stesso URI endpoint proxy, a condizione che abbiano valori di percorso proxy URI diversi.
- Un archivio di chiavi esterno con connettività pubblica agli endpoint non può utilizzare lo stesso valore di URI endpoint proxy di qualsiasi archivio di chiavi esterno con connettività ai servizi VPC endpoint nello stesso ambiente Regione AWS, anche se gli archivi di chiavi si trovano in archivi diversi. Account AWS
- Ogni archivio di chiavi esterno con connettività VPC endpoint deve avere il proprio nome privato. DNS L'URIendpoint proxy (DNSnome privato) deve essere univoco nella regione Account AWS and.

Percorso del proxy URI

Per creare un archivio di chiavi esterno, è necessario specificare il percorso di base nel proxy dell'archivio chiavi esterno verso il proxy richiesto APIs. Il valore deve iniziare con / e deve terminare con/kms/xks/v1, dove v1 rappresenta la versione del proxy AWS KMS API per l'archivio di chiavi esterne. 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'URIendpoint del proxy e del percorso del proxy. URI Ad esempio, se l'URIendpoint proxy è https://myproxy.xks.example.com e il URI percorso del proxy è/kms/xks/v1, AWS KMS invia le relative richieste proxy a. API https://myproxy.xks.example.com/kms/xks/v1

Puoi modificare il URI percorso del proxy, ma assicurati che il proxy dell'archivio chiavi esterno abbia accesso al materiale chiave associato alle KMS chiavi nell'archivio di chiavi esterno. In caso contrario, AWS KMS non è possibile decrittografare alcun testo cifrato crittografato con tali chiavi. KMS

Requisiti di unicità

- Il valore combinato del proxy URI endpoint (XksProxyUriEndpoint) e del URI percorso proxy (XksProxyUriPath) deve essere univoco nella regione and. Account AWS

VPCservizio endpoint

Speciifica il nome del servizio VPC endpoint Amazon utilizzato per comunicare con il tuo proxy di archiviazione chiavi esterno. Questo componente è richiesto solo per gli archivi di chiavi esterni che utilizzano la connettività del servizio VPC endpoint. Per informazioni sull'impostazione e la configurazione del servizio VPC endpoint per un archivio di chiavi esterno, consulta. Configura la VPC connettività del servizio endpoint

Il servizio VPC endpoint deve avere le seguenti proprietà:

- Il servizio VPC endpoint deve trovarsi nella stessa Account AWS area geografica dell'archivio di chiavi esterno.
- Deve disporre di un sistema di bilanciamento del carico di rete (NLB) connesso ad almeno due sottoreti, ognuna in una zona di disponibilità diversa.

- L'elenco dei principali consentiti per il servizio VPC endpoint deve includere l'entità del AWS KMS servizio per la regione:, ad esempio. `cks.kms.<region>.amazonaws.com` `cks.kms.us-east-1.amazonaws.com`
- Non deve richiedere l'accettazione delle richieste di connessione.
- Deve avere un DNS nome privato all'interno di un dominio pubblico di livello superiore. Ad esempio, potresti avere un DNS nome privato myproxy-private.xks.example.com nel dominio pubblico. `xks.example.com`

  Il DNS nome privato di un archivio di chiavi esterno con connettività al servizio endpoint deve essere univoco. VPC Regione AWS
- Lo stato di verifica del dominio del DNS nome privato deve essere`verified`.
- Il certificato del TLS server configurato sul proxy dell'archivio chiavi esterno deve specificare il DNS nome host privato presso il quale l'endpoint è raggiungibile.

Requisiti di unicità

- Gli archivi di chiavi esterni con connettività VPC endpoint possono condividerne uno`Amazon VPC`, ma ogni archivio di chiavi esterno deve avere il proprio servizio di VPC endpoint e il proprio nome privato. DNS

File di configurazione proxy

Un file di configurazione proxy è un file opzionale JSON che contiene i valori per il URIpercorso del proxy e le proprietà delle credenziali di autenticazione proxy dell'archivio di chiavi esterno. 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": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
  }
}
```

È possibile caricare un file di configurazione proxy solo quando si crea o si modifica un archivio di chiavi esterno nella AWS KMS console. Non è possibile utilizzarlo con le UpdateCustomKeyStoreoperazioni CreateCustomKeyStoreo, 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'CreateCustomKeyStoreoperazione.

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 create un archivio di chiavi esterno in AWS Management Console, potete caricare un file di configurazione proxy JSON basato sui valori relativi al URIpercorso del proxy e alle credenziali di autenticazione del proxy. Alcuni proxy generano automaticamente questo file, ma non è obbligatorio.

1. Accedi a AWS Management Console 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 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 di chiavi esterno.

7. Scegli o inserisci il nome del servizio VPC endpoint per questo archivio di chiavi esterno. Questo passaggio viene visualizzato solo quando il tipo di connettività proxy dell'archivio di chiavi esterno è il servizio VPCendpoint.

   Il servizio VPC endpoint e il suo VPCs devono soddisfare i requisiti per un archivio di chiavi esterno. Per informazioni dettagliate, consultare the section called “Assemblare i prerequisiti”.

8. Inserisci il tuo URIendpoint proxy. Il protocollo deve essereHTTPS. AWS KMS comunica sulla porta 443. Non specificare la porta nel valore dell'URIendpoint del proxy.

   Se AWS KMS riconosce il servizio VPC endpoint specificato nel passaggio precedente, completa questo campo automaticamente.

   Per la connettività pubblica degli endpoint, inserisci un endpoint disponibile pubblicamente. URI Per la connettività VPC degli endpoint, immettere `https://` seguito dal DNS nome privato del VPC servizio endpoint.

9. Per inserire i valori per il prefisso del URIpercorso del proxy e la credenziale di autenticazione del proxy, carica un file di configurazione del proxy o inserisci i valori manualmente.

   - Se disponi di un file di configurazione del proxy opzionale che contiene i valori per il URIpercorso del proxy e le credenziali di autenticazione del proxy, scegli 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 disponi di un file di configurazione del proxy, puoi inserire manualmente il URI percorso del proxy e i valori delle credenziali di autenticazione del proxy.

  a. Se non disponi di un file di configurazione del proxy, puoi inserire il proxy URI manualmente. La console fornisce il valore/kms/xks/v1 richiesto.

     Se il URIpercorso del proxy include un prefisso opzionale, ad esempio `example-prefix` in/*`example-prefix`*/`kms/xks/v1`, immetti il prefisso nel campo Prefisso del URIpercorso del 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.

Utilizzando il AWS KMS API

È possibile utilizzare l'CreateCustomKeyStoreoperazione 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 AWS KMS utilizzato per comunicare con il proxy di archiviazione delle chiavi esterno.
- `XksProxyUriPath`— Il percorso all'interno del proxy verso il proxyAPIs.
- `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 HTTPS valore HTTP o, ad esempio, il `XksProxyUriEndpoint` parametro.

```
aws configure set cli_follow_urlparam false
```

Altrimenti, la AWS CLI versione 1.0 sostituisce il valore del parametro con il contenuto trovato a quell'URIindirizzo, 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, sostituiscili 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 di chiavi esterno con connettività al servizio VPC endpoint.

```
$ 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.

```
{
    "CustomKeyStoreId": 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 l'archivio chiavi esterno è disconnesso, è possibile gestire l'archivio chiavi e KMS le relative chiavi, ma non è possibile creare o utilizzare KMS chiavi nell'archivio chiavi esterno. Per trovare lo stato di connessione dell'archivio di chiavi esterno, utilizza l'DescribeCustomKeyStoresoperazione 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 GetHealthStatusrichiesta 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 JSON basato sul quale sono specificati il URIpercorso del proxy e

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 di chiavi esterno perde definitivamente l'accesso alle chiavi esterne associate alle sue chiavi, il testo cifrato crittografato con tali KMS 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	✓	

Proprietà modificabili dell'archivio delle chiavi esterne	Qualsiasi stato di connessione	Richiede lo stato Disconnesso
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.	✓	
URIPercorso del proxy () XksProxyUriPath	✓	
Connettività proxy (XksProxyConnectivity) (È inoltre necessario aggiornare l'URIendpoint proxy. Se si passa alla connettività del servizio VPC endpoint, è necessario specificare un nome di servizio VPC endpoint proxy.)		✓
URIEndpoint proxy () XksProxyUriEndpoint Se si modifica l'endpoint proxyURI, potrebbe essere necessario modificare anche il certificato associato TLS.		✓
Nome del servizio VPC endpoint proxy () XksProxyVpcEndpointServiceName (Questo campo è obbligatorio per la connettività del servizio VPC endpoint)		✓

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'UpdateCustomKeyStoreoperazione.

Utilizzo della AWS KMS console

Quando modifichi un archivio delle chiavi esterne, puoi modificare qualsiasi valore configurabile. Alcune modifiche richiedono la disconnessione dell'archivio delle chiavi esterne dal relativo proxy.

Se stai modificando il URI percorso del proxy o la credenziale di autenticazione del proxy, puoi inserire i nuovi valori o caricare un file di configurazione del proxy dell'archivio chiavi esterno che includa i nuovi valori.

1. Accedi a AWS Management Console 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 modificare.
5. 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).
6. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Edit (Modifica).
7. Modifica una o più proprietà configurabili dell'archivio delle chiavi esterne. È inoltre possibile caricare un file di configurazione proxy dell'archivio chiavi esterno con i valori per il URI percorso del proxy e le credenziali di autenticazione del proxy. È possibile utilizzare un file di configurazione del proxy anche se alcuni valori specificati nel file non sono cambiati.
8. Scegli Update external key store (Aggiornamento dell'archivio delle chiavi esterne).
9. 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.
10. 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, Tuttavia, quando è disconnesso, non è possibile creare KMS chiavi nell'archivio chiavi esterno o utilizzare le chiavi nell'archivio KMS chiavi esterno nelle operazioni crittografiche.

Usando il AWS KMS API

Per modificare le proprietà di un archivio di chiavi esterno, utilizzare l'UpdateCustomKeyStoreoperazione. 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 di HTTP 200 e un JSON oggetto 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. È possibile lasciare l'archivio chiavi esterno nello stato disconnesso, ma è necessario ricollegarlo prima di poter creare nuove KMS chiavi nell'archivio chiavi o utilizzare KMS le chiavi esistenti nell'archivio chiavi per operazioni di crittografia.

> **Note**
>
> Se utilizzate la AWS CLI versione 1.0, eseguite il comando seguente prima di specificare un parametro con un HTTPS valore HTTP o, ad esempio, il parametro. `XksProxyUriEndpoint`
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> Altrimenti, la AWS CLI versione 1.0 sostituisce il valore del parametro con il contenuto trovato a quell'URIindirizzo, 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'UpdateCustomKeyStoreoperazione 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 la credenziale di autenticazione del proxy AWS KMS utilizzata per l'autenticazione sul proxy dell'archivio chiavi esterno. 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 `CustomKeyStoreId` 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
```

Cambia il percorso del proxy URI

L'esempio seguente aggiorna il URI percorso del proxy (`XksProxyUriPath`). La combinazione dell'URIendpoint del proxy e del URI percorso del proxy deve essere unica nella regione Account AWS 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 VPC servizio endpoint

L'esempio seguente utilizza l'UpdateCustomKeyStoreoperazione per modificare il tipo di connettività proxy dell'archivio chiavi esterno in`VPC_ENDPOINT_SERVICE`. Per apportare questa modifica, è necessario specificare i valori richiesti per la connettività del servizio VPC endpoint, incluso il nome del servizio VPC endpoint (`XksProxyVpcEndpointServiceName`) e un valore proxy URI endpoint (`XksProxyUriEndpoint`) che include il DNS nome privato del 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 del proxy URI endpoint (). `XksProxyUriEndpoint` Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

> **Note**
>
> VPCla connettività degli endpoint offre una maggiore sicurezza rispetto alla connettività degli endpoint pubblici. Prima di passare alla connettività pubblica degli endpoint, prendi in considerazione altre opzioni, tra cui l'individuazione del proxy di archiviazione delle chiavi esterno in sede e l'utilizzo del proxy solo per la VPC 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'DescribeCustomKeyStoresoperazione.

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 chiavi esterno, inclusi il tipo di connettività, l'URIendpoint e il percorso del proxy e l'ID della chiave di accesso della credenziale di autenticazione proxy corrente.
- Se il proxy dell'archivio chiavi esterno utilizza la connettività del servizio VPC endpoint, la console visualizza il nome del VPC servizio endpoint.
- 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. Tuttavia, se i tentativi di utilizzare una KMS chiave in un archivio di chiavi esterno connesso falliscono, ciò potrebbe indicare un problema con l'archivio chiavi esterno o il relativo proxy. 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, consultare 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'`XksProxyConfiguration`elemento della DescribeCustomKeyStoresrisposta. 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 di chiavi esterno utilizza la connettività degli endpoint pubblici o la connettività dei servizi VPC endpoint.

Endpoint proxy URI

L'endpoint AWS KMS utilizzato per connettersi al proxy di archiviazione delle chiavi esterno.

Percorso del proxy URI

Il percorso dall'URIendpoint proxy a cui AWS KMS invia APIle richieste 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

VPCnome del servizio endpoint

Il nome del servizio VPC endpoint Amazon che supporta l'archivio di chiavi esterno. Questo valore viene visualizzato solo quando l'archivio di chiavi esterno utilizza la connettività del servizio VPC endpoint. È possibile individuare il proxy dell'archivio chiavi esterno in VPC oppure utilizzare il servizio VPC endpoint per comunicare in modo sicuro con il proxy dell'archivio chiavi esterno.

Visualizza le proprietà del tuo archivio di chiavi esterno

È possibile visualizzare l'archivio di chiavi esterno e le proprietà associate nella AWS KMS console o utilizzando l'DescribeCustomKeyStoresoperazione.

Utilizzo della AWS KMS console

Per visualizzare gli archivi delle chiavi esterne in determinati account e regioni, utilizza la procedura seguente.

1. Accedi AWS Management Console 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.

Usando il AWS KMS API

Per visualizzare gli archivi di chiavi esterni, utilizzare l'DescribeCustomKeyStoresoperazione. 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 `CustomKeyStoreId` (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 include il tipo di connettività, l'URIendpoint proxy, il URIpercorso del proxy e l'ID della chiave di accesso della credenziale di autenticazione del 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 `CustomKeyStoreName` 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": [
    {
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleXksPublic",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-14T20:17:36.419000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://xks.example.com:6443",
        "UriPath": "/example/prefix/kms/xks/v1"
      }
    }
  ]
}
```

Il comando seguente ottiene un esempio di archivio di chiavi esterno con connettività al servizio VPC endpoint. 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. Tuttavia, se i tentativi

di utilizzare una KMS chiave in un key store esterno connesso falliscono, ciò potrebbe indicare un problema con il proxy dell'archivio chiavi esterno o con 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": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "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"
      }
    }
  ]
}
```

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. Puoi modificare il fuso orario in UTCo 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 KMS chiavi 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 AWS Support 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 una credenziale errata e il proxy risponde con una`AuthenticationFailedException`, il grafico indicherà comunque un'affidabilità del 100% perché il proxy ha identificato un valore errato nella APIrichiesta proxy dell'archivio chiavi esterno e pertanto è previsto l'errore. 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 si aspetta che il proxy dell'archivio chiavi esterno risponda a ogni 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 tuo gestore di chiavi esterno non è in grado di gestire la frequenza di 1800 richieste al secondo, prendi in considerazione la possibilità di richiedere una riduzione della quota di richieste di KMS chiavi in un archivio di chiavi personalizzato. Le richieste di operazioni crittografiche che utilizzano le KMS chiavi dell'archivio di chiavi esterno falliranno rapidamente con un'eccezione di limitazione, anziché essere elaborate e successivamente rifiutate dal proxy dell'archivio chiavi esterno o dal gestore di chiavi esterno.

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 picco `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 visualizzate. 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 `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.
Irreversibile	**`AuthenticationFailedException`** Per la risoluzione dei problemi, consultare Errori delle credenziali di autenticazione.	**`XksProxyIncorrectAuthenticationCredentialException`** in risposta alle operazioni `CreateCustomKeyStore` e `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** in risposta alle operazioni `CreateKey` . **`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 `CreateCustomKeyStore` e `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`**

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
		in risposta alle operazioni `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.
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 `CreateCustomKeyStore` e `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** in risposta alle operazioni `CreateKey` . **`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.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	**`InvalidKeyUsageException`** Per la risoluzione dei problemi, consultare Errori relativi alle operazioni di crittografia per la chiave esterna.	**`XksKeyInvalidConfigurationException`** in risposta alle operazioni `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.
Irreversibile	**`InvalidStateException`** Per la risoluzione dei problemi, consultare Errori relativi alle operazioni di crittografia per la chiave esterna.	**`XksKeyInvalidConfigurationException`** in risposta alle operazioni `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.
Irreversibile	**`InvalidUriPathException`** Per la risoluzione dei problemi, consultare Errori di configurazione generale.	**`XksProxyInvalidConfigurationException`** in risposta alle operazioni `CreateCustomKeyStore` e `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** in risposta alle operazioni `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	**`KeyNotFoundException`** Per la risoluzione dei problemi, consultare Errori relativi alla chiave esterna.	**`XksKeyNotFoundException`** in risposta alle operazioni `CreateKey` . **`KMSInvalidStateException`** in risposta alle operazioni di crittografia.
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 i KMS tasti in questo archivio di chiavi esterno.	**`XksProxyUriUnreachableException`** in risposta alle operazioni `CreateCustomKeyStore` e `UpdateCustomKeyStore` . **`CustomKeyStoreInvalidStateException`** in risposta alle operazioni `CreateKey` . **`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 `CreateKey` . **`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 visualizzi ripetutamente questo errore, verifica che il proxy dell'archivio chiavi esterno sia attivo e connesso alla rete e che il URI percorso e il nome dell'endpoint URI o del VPC servizio siano corretti nell'archivio chiavi esterno.	**`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 irreparabili 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 TLS certificato per l'endpoint proxy dell'archivio chiavi esterno (). `XksProxyUriEndpoint` Utilizzate questo grafico per monitorare la scadenza imminente del certificato. TLS

Quando il certificato scade, AWS KMS non è possibile comunicare con il proxy dell'archivio chiavi esterno. Tutti i dati protetti da KMS chiavi nell'archivio delle chiavi esterno 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 un archivio di chiavi esterno, è necessario collegare l'archivio di 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, è possibile visualizzare e gestire l'archivio di chiavi esterno e le relative KMS chiavi esistenti. Tuttavia, non è possibile creare KMS chiavi nell'archivio di KMS chiavi esterno o utilizzarle nelle operazioni crittografiche. 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 ne utilizzano le 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 si disconnette un archivio chiavi personalizzato, KMS le chiavi nell'archivio chiavi diventano immediatamente inutilizzabili (a seconda dell'eventuale coerenza). Tuttavia, le risorse crittografate con chiavi dati protette dalla KMS chiave non vengono influenzate fino a quando la KMS chiave non viene riutilizzata, 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 inutilizzabili influiscono sulle chiavi dati KMS.

> **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

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'DescribeCustomKeyStoresoperazione o la AWS KMS console. Lo stato della connessione viene visualizzato in ogni tabella dell'archivio di chiavi personalizzato, nella sezione Configurazione generale della pagina di dettaglio di ogni archivio di chiavi personalizzato e nella scheda Configurazione crittografica delle KMS chiavi in un archivio di chiavi personalizzato. 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. È possibile creare e utilizzare KMS chiavi nell'archivio chiavi personalizzato.

  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.

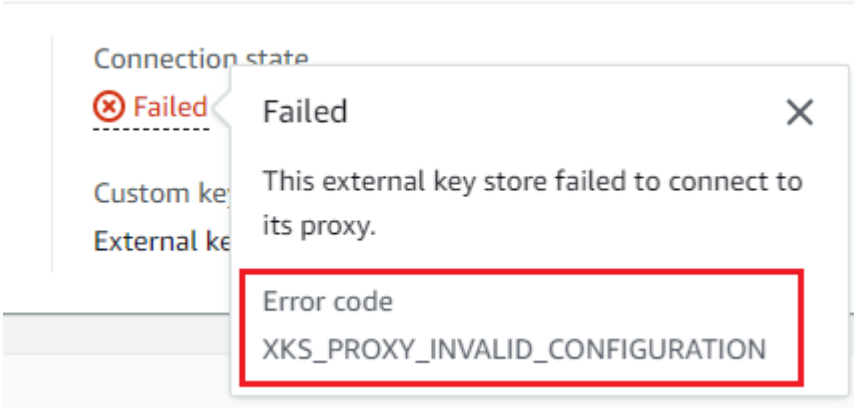
  Uno CONNECTED stato indica che una connessione è riuscita e che l'archivio chiavi personalizzato non è stato disconnesso intenzionalmente. Non indica che la connessione sta funzionando correttamente. Per informazioni sullo stato del AWS CloudHSM cluster associato all'archivio delle AWS CloudHSM chiavi, consulta Ottenere le CloudWatch metriche AWS CloudHSM nella Guida per l'utente. AWS CloudHSM 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 DescribeCustomKeyStoresrisposta 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 DescribeCustomKeyStoresrisposta, 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 chiavi esterno è collegato al relativo proxy di archiviazione chiavi esterno, è possibile creare KMS chiavi nell'archivio chiavi esterno e utilizzare le KMS chiavi esistenti nelle operazioni crittografiche.

Il processo che collega un archivio delle chiavi esterne al relativo proxy varia in base alla connettività dell'archivio.

- Quando si connette un archivio di chiavi esterno con connettività endpoint pubblica, AWS KMS invia una GetHealthStatus richiesta al proxy dell'archivio chiavi esterno per convalidare l'URIendpoint proxy, il URI percorso del proxye le credenziali di autenticazione del proxy. Una risposta corretta del proxy conferma che l'URIendpoint e il URI percorso del proxy sono accurati e accessibili e che il proxy ha autenticato la richiesta firmata con le credenziali di autenticazione proxy per l'archivio di chiavi esterno.

- Quando si collega un archivio di chiavi esterno con connettività del servizio VPC endpoint al relativo proxy di archiviazione chiavi esterno, AWS KMS effettua le seguenti operazioni:
  - Conferma che il dominio per il DNS nome privato specificato nell'URIendpoint proxy è verificato.
  - Crea un endpoint di interfaccia da un servizio AWS KMS VPC all'VPCendpoint dell'utente.
  - Crea una zona ospitata privata per il DNS nome privato specificato nell'endpoint proxy URI
  - Invia una GetHealthStatusrichiesta al proxy dell'archivio chiavi esterno. Una risposta corretta del proxy conferma che l'URIendpoint e il URIpercorso del proxy sono accurati e accessibili e che il proxy ha autenticato la richiesta firmata con la credenziale di autenticazione del proxy per l'archivio di chiavi esterno.

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'DescribeCustomKeyStoresoperazione 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'ConnectCustomKeyStoreoperazione.

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 AWS Management Console 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 lo stato di connessione dell'archivio chiavi esterno è FAILED, è necessario disconnettere l'archivio chiavi esterno prima di collegarlo.

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 della connessione cambia in. CONNECTED

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.

Utilizzo del AWS KMS API

Per connettere un archivio di chiavi esterno disconnesso, utilizzare l'ConnectCustomKeyStoreoperazione.

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. A meno che non fallisca rapidamente, `ConnectCustomKeyStore` restituisce una risposta di HTTP 200 e un JSON oggetto senza proprietà. Questa risposta iniziale non indica tuttavia che la connessione è riuscita. Per determinare se l'archivio chiavi esterno è connesso, consultate lo stato della connessione nella DescribeCustomKeyStoresrisposta.

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'DescribeCustomKeyStoresoperazione. 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'DescribeCustomKeyStoresoperazione. 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 VPC endpoint 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 VPC endpoint Amazon e che il servizio VPC endpoint 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 si disconnette un archivio di chiavi esterno con connettività del servizio VPC endpoint dal relativo proxy di archiviazione chiavi esterno, AWS KMS elimina l'endpoint di interfaccia con il servizio VPC endpoint e rimuove l'infrastruttura di rete creata per supportare la connessione. Non è richiesto alcun processo equivalente per gli archivi delle chiavi esterne con connettività dell'endpoint pubblico. Questa azione non influisce sul servizio VPC endpoint o sui relativi componenti di supporto e non influisce sul proxy dell'archivio chiavi esterno o sui componenti esterni.

Sebbene l'archivio chiavi esterno sia 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 KMS chiavi nell'archivio chiavi esterno disconnesso si trovano in uno stato `UNAVAILABLE` chiave (a meno che non siano in attesa di eliminazione), il che significa che non possono essere utilizzate nelle operazioni crittografiche. Tuttavia, è ancora possibile visualizzare e gestire l'archivio di chiavi esterno e le relative chiavi esistenti. KMS

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**
>
> Anche se un archivio chiavi personalizzato è disconnesso, tutti i tentativi di creare KMS chiavi nell'archivio chiavi personalizzato o di utilizzare KMS le chiavi esistenti nelle operazioni crittografiche falliranno. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Per stimare meglio l'effetto della disconnessione dell'archivio chiavi esterno, identificate KMS le chiavi nell'archivio chiavi esterno e determinatene l'uso passato.

Puoi disconnettere un archivio delle chiavi esterne per i seguenti motivi:

- Per modificarne le proprietà. È possibile modificare il nome dell'archivio chiavi personalizzato, il URI percorso proxy e la credenziale di autenticazione proxy mentre l'archivio chiavi esterno è connesso. Tuttavia, per modificare il tipo di connettività proxy, l'URIendpoint proxy o il nome del servizio dell'VPCendpoint, è necessario prima disconnettere l'archivio di chiavi esterno. 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 VPC il servizio endpoint. 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 KMS le chiavi nell'archivio chiavi esterno. È possibile disabilitare e riattivare KMS le chiavi in un archivio di chiavi esterno utilizzando la AWS KMS console o l'DisableKeyoperazione. Queste operazioni vengono completate rapidamente (a seconda dell'eventuale coerenza), ma agiscono su una KMS chiave alla volta. La disconnessione dell'archivio chiavi esterno modifica lo stato delle chiavi di tutte le KMS chiavi nell'archivio chiavi esterno`Unavailable`, il che impedisce che vengano utilizzate in qualsiasi operazione crittografica.
- 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'DisconnectCustomKeyStoreoperazione.

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 AWS Management Console 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 termine dell'operazione, lo stato della connessione cambia da a. DISCONNECTINGDISCONNECTED 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.

Usando il AWS KMS API

Per disconnettere un archivio di chiavi esterno connesso, utilizzare l'DisconnectCustomKeyStoreoperazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta di HTTP 200 e un JSON oggetto senza proprietà. Il completamento del processo può richiedere fino a cinque minuti. Per trovare lo stato di connessione dell'archivio chiavi esterno, utilizzare l'DescribeCustomKeyStoresoperazione.

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 chiavi esterno con connettività al servizio VPC endpoint. 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 di chiavi esterno sia disconnesso, utilizzare l'operazione. DescribeCustomKeyStores 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 di chiavi esterno

Quando si elimina un archivio 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 Amazon VPC o un servizio VPC endpoint.

Prima di eliminare un archivio di chiavi esterno, devi eliminare tutte le KMS chiavi dall'archivio chiavi e disconnettere l'archivio chiavi dal relativo proxy esterno. 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 di crittografia simmetriche nell'archivio KMS chiavi esterno, anche se si ha accesso allo stesso materiale per le chiavi esterne. AWS KMS include i metadati nel testo cifrato simmetrico unico per ogni chiave. KMS Questa funzionalità di sicurezza garantisce che solo la KMS chiave che ha crittografato i dati possa decrittografarli.

Anziché eliminare l'archivio delle chiavi esterne, puoi disconnetterlo. Quando un archivio chiavi esterno è disconnesso, è possibile gestire l'archivio chiavi esterno e il relativo archivio, AWS KMS keys ma non è possibile creare o utilizzare KMS le chiavi nell'archivio chiavi esterno. È possibile ricollegare l'archivio di chiavi esterno in qualsiasi momento e riprendere a utilizzarne KMS le chiavi per crittografare e decrittografare i dati. Non è previsto alcun costo per un proxy di archiviazione chiavi esterno disconnesso o per le relative chiavi non disponibili. KMS

È possibile eliminare l'archivio di chiavi esterno nella AWS KMS console o utilizzando l'DeleteCustomKeyStoreoperazione.

Utilizzo della AWS KMS console

È possibile utilizzare la AWS KMS console per eliminare un archivio di chiavi esterno.

1. Accedi AWS Management Console 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 lo stato di connessione dell'archivio chiavi esterno non lo è DISCONNECTED, è necessario disconnettere l'archivio chiavi esterno 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.

Usando il AWS KMS API

Per eliminare un archivio di chiavi esterno, utilizzare l'DeleteCustomKeyStoreoperazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta di HTTP 200 e un JSON oggetto 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'DeleteCustomKeyStoreoperazione 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'DescribeCustomKeyStoresoperazione.

```
$ 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 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 fallisce. 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, ad esempio problemi con la configurazione o il funzionamento del proxy dell'archivio chiavi esterno, del gestore di chiavi esterno, delle chiavi esterne o del servizio VPC endpoint.
- Rete: potrebbe essere un problema di connettività o di rete, ad esempio un problema con l'endpoint proxy, la porta o il DNS nome o il dominio privato.

> **Note**
>
> Quando le operazioni di gestione negli archivi delle chiavi esterne hanno esito negativo, generano diverse eccezioni. Tuttavia, le operazioni AWS KMS crittografiche restituiscono `KMSInvalidStateException` risultati 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'ConnectCustomKeyStoreoperazione 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

- 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 chiavi esterno e il gestore di chiavi esterno potrebbero fornire metodi più semplici per creare e gestire l'archivio chiavi esterno e KMS le relative chiavi. 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. È possibile trovare ulteriori informazioni sulle AWS KMS eccezioni nella sezione Reference and.AWS Key Management Service API 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 ripresenta in un AWS servizio diverso, ad esempio quando si utilizza una KMS chiave 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 KMS chiave.

CloudTrail registri

Ogni AWS KMS API operazione, incluse le azioni nella AWS KMS console, viene registrata nei AWS CloudTrail registri. 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 Decifra l'errore con una KMS chiave in un archivio di chiavi esterno.

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 TLS certificato 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.

Visualizzazioni di archivi e chiavi esterni KMS

AWS KMS visualizza informazioni dettagliate sugli archivi di KMS chiavi esterni e sulle chiavi nell'archivio chiavi esterno della AWS KMS console e sulla risposta alle DescribeKeyoperazioni

DescribeCustomKeyStoresand. Queste visualizzazioni includono campi speciali per gli archivi di KMS chiavi esterni e le chiavi con informazioni che è possibile utilizzare per la risoluzione dei problemi, come lo stato della connessione dell'archivio chiavi esterno e l'ID della chiave esterna associata alla KMS chiave. Per informazioni dettagliate, consultare Visualizza gli archivi di chiavi esterni.

XKSProxy Test Client

AWS KMS fornisce un client di test open source che verifica che il proxy dell'archivio chiavi esterno sia conforme alla specifica del proxy AWS KMS External Key Store. API Puoi utilizzare tale client di test per identificare e risolvere i problemi relativi al proxy dell'archivio delle chiavi esterne.

Errori di configurazione

Quando si crea un archivio di chiavi esterno, si specificano i valori delle proprietà che comprendono la configurazione dell'archivio di chiavi esterno, come la credenziale di autenticazione del proxy, l'endpoint proxy, il URIpercorso del proxy e il nome del servizio dell'URIendpoint. 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. È possibile correggere un URI percorso proxy o una credenziale di autenticazione proxy non validi senza disconnettere l'archivio di chiavi esterno. 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 URI percorso del proxy e i valori delle credenziali di autenticazione proxy, durante la creazione o l'aggiornamento dell'archivio di chiavi esterno, carica un file di configurazione del proxy sulla console. AWS KMS Si tratta di un file JSON basato sul URI percorso del proxy e sui valori delle credenziali di autenticazione proxy fornito dal proxy dell'archivio chiavi esterno o dal gestore delle chiavi esterno. Non è possibile utilizzare un file di configurazione proxy per le AWS KMS API operazioni, ma è possibile utilizzare i valori del file per fornire valori dei parametri per le API richieste che corrispondono 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 agli endpoint, AWS KMS verifica i valori delle proprietà quando crei e aggiorni l'archivio di chiavi esterno. Per gli archivi di chiavi esterni con connettività al servizio VPC endpoint, AWS KMS verifica i valori delle proprietà quando ti connetti e aggiorni l'archivio di chiavi esterno.

> **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 chiavi esterno ha rifiutato la richiesta a causa di un percorso non validoURI. Verifica il URI percorso del tuo archivio di chiavi esterno e aggiornalo se necessario.

- Il URIpercorso del proxy è il percorso di base per AWS KMS le richieste al proxyAPIs. Se questo percorso non è corretto, tutte le richieste al proxy hanno esito negativo. Per visualizzare il URI percorso proxy corrente per il tuo archivio di chiavi esterno, usa la AWS KMS console o l'`DescribeCustomKeyStores`operazione. Per trovare il URI percorso proxy corretto, consulta la documentazione del proxy dell'archivio chiavi esterno. Per informazioni sulla correzione del valore del URI percorso del proxy, consultaModifica delle proprietà dell'archivio chiavi esterno.
- Il URI percorso del proxy per il proxy dell'archivio chiavi esterno può cambiare con gli aggiornamenti del proxy dell'archivio chiavi esterno o del gestore di chiavi esterno. 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 possibile stabilire una TLS connessione al proxy dell'archivio chiavi esterno. Verifica la TLS configurazione, incluso il relativo certificato.

- Tutti i proxy di archiviazione delle chiavi esterni richiedono un TLS certificato. Il TLS certificato deve essere emesso da un'autorità di certificazione (CA) pubblica supportata per gli archivi di chiavi esterni. Per un elenco di quelli supportatiCAs, consulta Trusted Certificate Authorities nella AWS KMS External Key Store Proxy API Specification.
- Per la connettività pubblica degli endpoint, il nome comune dell'oggetto (CN) sul TLS certificato deve corrispondere al nome di dominio nell'URIendpoint proxy per il proxy dell'archivio chiavi esterno. Ad esempio, se l'endpoint pubblico è https://myproxy.xks.example.com, ilTLS, il CN sul TLS certificato deve essere o. `myproxy.xks.example.com` `*.xks.example.com`
- Per la connettività del servizio VPC endpoint, il nome comune dell'oggetto (CN) sul TLS certificato deve corrispondere al DNS nome privato del servizio VPCendpoint. Ad esempio, se il DNS nome privato è myproxy-private.xks.example.com, il CN sul certificato deve essere o. TLS `myproxy-private.xks.example.com` `*.xks.example.com`
- Il certificato non può essere scaduto. TLS Per ottenere la data di scadenza di un TLS certificato, utilizza SSL strumenti come Open SSL. Per monitorare la data di scadenza di un TLS certificato associato a un archivio di chiavi esterno, utilizza la XksProxyCertificateDaysToExpire CloudWatch metrica. Il numero di giorni che mancano alla data di scadenza della TLS certificazione viene visualizzato anche nella sezione Monitoraggio della AWS KMS console.
- Se utilizzi la connettività pubblica degli endpoint, utilizza gli strumenti SSL di test per testare la SSL configurazione. TLSgli errori di connessione possono derivare da un concatenamento errato dei certificati.

VPCerrori di configurazione della connettività del servizio endpoint

Eccezioni: `XksProxyVpcEndpointServiceNotFoundException`, `XksProxyVpcEndpointServiceInvalidConfigurationException`

Oltre ai problemi di connettività generali, è possibile riscontrare i seguenti problemi durante la creazione, la connessione o l'aggiornamento di un archivio di chiavi esterno con connettività del servizio VPC endpoint. AWS KMS verifica i valori delle proprietà di un archivio di chiavi esterno con connettività al servizio VPC endpoint durante la creazione, la connessione e l'aggiornamento

dell'archivio di 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 di servizio VPC endpoint errato. Verificare che il nome del servizio VPC endpoint per l'archivio di chiavi esterno sia corretto e corrisponda al valore dell'URIendpoint proxy per l'archivio di chiavi esterno. Per trovare il nome del servizio VPC endpoint, usa la VPCconsole Amazon o l'DescribeVpcEndpointServicesoperazione. Per trovare il nome del servizio VPC endpoint e l'URIendpoint proxy di un archivio di chiavi esterno esistente, usa la AWS KMS console o l'operazione. DescribeCustomKeyStores Per informazioni dettagliate, consultare Visualizza gli archivi di chiavi esterni.
- Il servizio VPC endpoint potrebbe trovarsi in un archivio di chiavi Regione AWS diverso da quello esterno. Verifica che il servizio VPC endpoint e l'archivio chiavi esterno si trovino nella stessa regione. (Il nome esterno del nome della regione, ad esempio, fa parte del nome del servizio VPC endpoint`us-east-1`, ad esempio com.amazonaws.vpce.us-east-1. vpce-svc-example.) Per un elenco dei requisiti per il servizio VPC endpoint per un archivio di chiavi esterno, vedereVPCservizio endpoint. Non è possibile spostare un servizio VPC endpoint o un archivio di chiavi esterno in un'altra regione. Tuttavia, è possibile creare un nuovo archivio di chiavi esterno nella stessa regione del servizio VPC endpoint. Per informazioni dettagliate, consulta Configura la VPC connettività del servizio endpoint e Creare un archivio di chiavi esterno.
- AWS KMS non è un principale consentito per il servizio VPC endpoint. L'elenco dei principali consentiti per il servizio VPC endpoint deve includere il `cks.kms.<region>.amazonaws.com` valore, 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 VPC endpoint non soddisfa uno dei seguenti requisiti:

- VPCRichiede almeno due sottoreti private, ciascuna in una zona di disponibilità diversa. Per assistenza nell'aggiunta di una sottorete alla tuaVPC, consulta Create a subnet in your VPC nella Amazon VPC User Guide.
- Il tipo di servizio VPC endpoint deve utilizzare un sistema di bilanciamento del carico di rete, non un sistema di bilanciamento del carico gateway.
- L'accettazione non deve essere richiesta per il servizio VPC endpoint (l'accettazione richiesta deve essere falsa). Se è richiesta l'accettazione manuale di ogni richiesta di connessione, AWS KMS non è possibile utilizzare il servizio VPC endpoint per connettersi al proxy esterno dell'archivio chiavi. Per maggiori dettagli, consulta Accettare o rifiutare le richieste di connessione nella Guida di AWS PrivateLink .
- Il servizio VPC endpoint deve avere un DNS nome privato che sia un sottodominio di un dominio pubblico. Ad esempio, se il DNS nome privato è`https://myproxy-private.xks.example.com`, i `example.com` domini `xks.example.com` o devono avere un server pubblico. DNS Per visualizzare o modificare il DNS nome privato del servizio VPC endpoint, consulta Gestire DNS i nomi per i servizi VPC endpoint nella Guida.AWS PrivateLink
- Lo stato di verifica del dominio per il tuo DNS nome privato deve essere. `verified` Per visualizzare e aggiornare lo stato di verifica del dominio con DNS nome privato, consultaPassaggio 5: Verifica il tuo DNS nome di dominio 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 DNS dominio privato può essere verificato solo se è il sottodominio di un dominio pubblico. In caso contrario, lo stato di verifica del DNS dominio privato non cambia, anche dopo aver aggiunto il TXT record richiesto.

- Il DNS nome privato del servizio VPC endpoint deve corrispondere al valore dell'URIendpoint proxy per l'archivio di chiavi esterno. Per un archivio di chiavi esterno con connettività VPC al servizio endpoint, l'URIendpoint proxy deve essere `https://` seguito dal DNS nome privato del servizio endpoint. VPC Per visualizzare il valore dell'URIendpoint proxy, vedere. Visualizza gli archivi di chiavi esterni Per modificare il valore dell'URIendpoint del proxy, vedere. 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. A meno che non fallisca rapidamente, l'`ConnectCustomKeyStore`operazione restituisce una risposta di HTTP 200 e un JSON oggetto 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`.

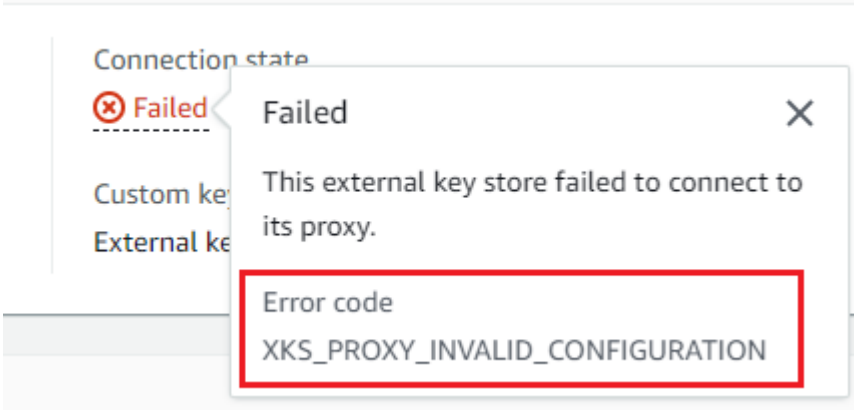
Per visualizzare lo stato di connessione di un archivio delle chiavi esterne:

- Nella DescribeCustomKeyStoresrisposta, 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 DescribeCustomKeyStoresrisposta, 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 non è stato possibile 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 TLS configurazione non è valida. Verificare che il proxy dell'archivio chiavi esterno supporti TLS 1.2 o 1.3. Verifica inoltre che il TLS certificato non sia scaduto, che corrisponda al nome host indicato nel `XksProxyUriEndpoint` valore e che sia firmato da un'autorità di certificazione attendibile 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 VPC endpoint Amazon non è conforme ai requisiti per un archivio di chiavi AWS KMS esterno.

- Il servizio VPC endpoint deve essere un servizio endpoint per gli endpoint di interfaccia del chiamante. Account AWS
- Deve disporre di un sistema di bilanciamento del carico di rete (NLB) connesso ad almeno due sottoreti, ognuna in una zona di disponibilità diversa.
- L'`Allow principals`elenco deve includere il principale AWS KMS servizio 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 DNS nome privato. Il DNS nome privato di un archivio di chiavi esterno con `VPC_ENDPOINT_SERVICE` connettività deve essere univoco Regione AWS.
- Il dominio del DNS nome privato deve avere lo stato di verifica di`verified`.
- Il TLScertificato specifica il DNS nome host privato presso il quale l'endpoint è raggiungibile.

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

AWS KMS non riesce a trovare il servizio VPC endpoint che utilizza per comunicare con il proxy di archiviazione delle chiavi esterno. Verifica che `XksProxyVpcEndpointServiceName` sia corretto e che il responsabile del AWS KMS servizio disponga delle autorizzazioni di consumatore del servizio sul servizio VPC endpoint Amazon.

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. `CreateCustomKeyStore``XksProxyUriUnreachableException` 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 chiavi esterno sia attivo e connesso alla rete e che l'URIendpoint proxy, il URI percorso del proxy e il nome del servizio dell'VPCendpoint (se disponibile) siano corretti nell'archivio chiavi esterno. Inoltre, verifica che il gestore delle chiavi esterno sia vicino a quello dell'archivio Regione AWS di 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 ripetutamente questo errore, verifica che il proxy dell'archivio chiavi esterno sia attivo e connesso alla rete e che il relativo endpoint URI sia corretto nell'archivio chiavi esterno.

- Il proxy dell'archivio chiavi esterno non ha risposto a una API richiesta 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'ConnectCustomKeyStoreoperazione fallisce, 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 è in grado di comunicare con il proxy dell'archivio chiavi esterno. Verifica che il proxy dell'archivio chiavi esterno sia attivo e connesso alla rete e che il URI percorso e il nome dell'endpoint URI o del VPC servizio siano corretti nell'archivio di chiavi esterno.

Questo errore può verificarsi per i seguenti motivi:

- Il proxy dell'archivio delle chiavi esterne non è attivo o non è connesso alla rete.
- È presente un errore nei valori dell'URIendpoint proxy, del URIpercorso del proxy o del nome del servizio dell'VPCendpoint (se applicabile) nella configurazione dell'archivio

chiavi esterno. Per visualizzare la configurazione dell'archivio chiavi esterno, utilizza l'DescribeCustomKeyStoresoperazione 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. Questo valore non è configurabile.
- Quando il proxy dell'archivio chiavi esterno segnala (in una GetHealthStatusrisposta) che tutte le istanze del gestore di chiavi esterno lo sonoUNAVAILABLE, l'ConnectCustomKeyStoreoperazione 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 delle 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'ammin istratore 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 di chiavi esterno e Amazon VPC per l'archivio chiavi esterno non funziona correttamente. Se lo utilizzi AWS Direct Connect, verifica che il tuo gestore di chiavi esterno VPC sia in grado di comunicare in modo efficace. Per assistenza nella risoluzione di eventuali problemi, consulta Risoluzione dei problemi AWS Direct Connect nella Guida per l' AWS Direct Connect utente.

`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 HTTP intestazioni 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 negli archivi di chiavi personalizzati. KMS

- Quando un'operazione di gestione, ad esempio`CancelKeyDeletion`, fallisce e restituisce questa eccezione, indica che lo stato della KMS chiave non è compatibile con l'operazione.
- Quando un'operazione di crittografia su una KMS chiave in un archivio di chiavi personalizzato ha esito negativo`KMSInvalidStateException`, può indicare un problema relativo

allo stato della KMS chiave. Tuttavia, l'operazione AWS KMS crittografica restituisce tutti `KMSInvalidStateException` gli errori di configurazione esterni e 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 un' AWS KMS APIoperazione, vedereStati chiave delle AWS KMS chiavi. Per trovare lo stato della KMS chiave, nella pagina Chiavi gestite dal cliente, visualizza il campo Stato della KMS chiave. Oppure, utilizza l'DescribeKeyoperazione e visualizza l'`KeyState`elemento nella risposta. Per informazioni dettagliate, consultare Identifica e visualizza le chiavi.

> **Note**
>
> Lo stato della KMS chiave in un archivio di chiavi esterno non indica nulla sullo 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 chiavi esterno, non allo stato della chiave di una KMS chiave.

Un'operazione di crittografia su una KMS chiave in un archivio personalizzato potrebbe non riuscire perché lo stato della KMS chiave è `Unavailable` o`PendingDeletion`. (I tasti disattivati restituiscono `DisabledException`).

- Una KMS chiave ha uno stato `Disabled` chiave solo quando la KMS si disattiva intenzionalmente nella AWS KMS console o utilizzando l'DisableKeyoperazione. Quando una KMS chiave è disattivata, è possibile visualizzarla e gestirla, ma non utilizzarla nelle operazioni crittografiche. Per risolvere il problema, abilita la chiave. Per informazioni dettagliate, consultare Attivazione e disattivazione dei tasti.
- Una KMS chiave ha uno stato `Unavailable` chiave quando l'archivio chiavi esterno viene disconnesso dal proxy dell'archivio chiavi esterno. Per correggere una KMS chiave non disponibile, ricollega l'archivio chiavi esterno. Dopo la riconnessione dell'archivio chiavi esterno, lo stato delle KMS chiavi nell'archivio chiavi esterno viene automaticamente ripristinato allo stato precedente, ad `Enabled` esempio o. `Disabled`

  Una KMS chiave ha uno stato `PendingDeletion` chiave quando è stata pianificata per l'eliminazione ed è nel periodo di attesa. Un errore di stato della KMS chiave su una chiave 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 KMS chiave, annulla l'eliminazione pianificata, quindi abilita la chiave. Per informazioni dettagliate, consultare Pianifica l'eliminazione della chiave.

Errori di decrittografia

Eccezioni: `KMSInvalidStateException`

Quando un'operazione di decrittografia con una KMS chiave in un archivio 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 KMS chiave 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` richiesta 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 presente nel gestore di chiavi esterno che funge da materiale per la chiave esterna. KMS AWS KMS non può 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 di chiavi esterno quando si crea una KMS chiave nell'archivio delle chiavi esterno. Non è possibile modificare l'ID della chiave esterna dopo la creazione della KMS chiave. Per evitare problemi con la KMS chiave, l'`CreateKey`operazione richiede al proxy dell'archivio chiavi esterno di verificare l'ID e la configurazione della chiave esterna. Se la chiave esterna non soddisfa i requisiti per l'utilizzo con una KMS chiave, l'`CreateKey`operazione ha esito negativo e viene visualizzata un'eccezione e un messaggio di errore che identifica il problema.

Tuttavia, possono verificarsi problemi dopo la creazione della KMS chiave. 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'CreateKeyoperazione tenta di verificare l'ID e le proprietà della chiave esterna fornita nel parametro External key ID (console) o `XksKeyId` (API). Questa pratica è progettata per rilevare gli errori prima di provare a utilizzare la chiave esterna con la KMS chiave.

Chiave esterna in uso

Ogni KMS chiave in un archivio di chiavi esterno deve utilizzare una chiave esterna diversa. Quando `CreateKey` riconosce che l'ID della chiave esterna (XksKeyId) per una KMS chiave non è univoco nell'archivio chiavi esterno, fallisce con un`XksKeyAlreadyInUseException`.

Se ne usi più di una IDs per la stessa chiave esterna, `CreateKey` non riconoscerà la chiave duplicata. Tuttavia, KMS le chiavi 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 KMS chiave, l'`CreateKey`operazione ha esito negativo e viene restituito il seguente `XksKeyNotFoundException` messaggio 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 KMS chiave 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, utilizzate una chiave esterna diversa con la KMS chiave. Per un elenco dei requisiti per la chiave esterna, consulta Requisiti per una KMS chiave in un archivio di chiavi esterno.

Requisiti della chiave esterna non soddisfatti

Quando il proxy dell'archivio chiavi esterno segnala che la chiave esterna non soddisfa i requisiti per l'utilizzo con una KMS chiave, l'`CreateKey`operazione ha esito negativo e viene `XksKeyInvalidConfigurationException` restituito uno dei seguenti messaggi di errore.

La specifica chiave della chiave esterna deve essere AES _256. La specifica chiave della chiave esterna specificata è *<key-spec>* .

- La chiave esterna deve essere una chiave di crittografia simmetrica a 256 bit con una specifica di chiave di _256. AES 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 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 esterna deve includere ENCRYPT eDECRYPT. 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 chiavi esterno non riesce a trovare la chiave esterna associata alla KMS chiave o se la chiave esterna non soddisfa i requisiti per l'utilizzo con una KMS chiave, l'operazione di crittografia ha esito negativo.

I problemi relativi alle chiavi esterne rilevati durante un'operazione di crittografia sono più difficili da risolvere rispetto ai problemi di chiave esterna rilevati prima della creazione della KMS chiave. Non è possibile modificare l'ID della chiave esterna dopo la creazione della KMS chiave. Se la KMS chiave non ha ancora crittografato alcun dato, puoi eliminare la KMS chiave e crearne una nuova con un

ID di chiave esterno diverso. Tuttavia, il testo cifrato generato con la KMS chiave non può essere decrittografato da nessun'altra KMS chiave, 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 KMS chiave, le operazioni crittografiche restituiscono un `KMSInvalidStateException` messaggio di errore con il seguente messaggio 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 KMS chiave non è più valido.

  Per trovare l'ID della chiave esterna associata alla tua KMS chiave, visualizza i dettagli della KMS chiave. 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 KMS chiave 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 è possibile modificare l'ID della chiave esterna associata a una KMS chiave esistente. Per decrittografare qualsiasi testo cifrato crittografato sotto la KMS chiave, è necessario riassociare la chiave esterna all'ID della chiave esterna esistente.

  Se non hai ancora utilizzato la KMS chiave per crittografare i dati, puoi creare una nuova KMS chiave con un ID di chiave esterna valido. Tuttavia, se avete generato testo cifrato con la KMS chiave, non potete usare nessun'altra KMS chiave per decrittografare il testo cifrato, anche se si 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 cifrato crittografato con la chiave associata non è recuperabile. KMS

Errori di configurazione della chiave esterna

Quando il proxy dell'archivio chiavi esterno segnala che la chiave esterna non soddisfa i requisiti per l'utilizzo con una KMS chiave, l'operazione crittografica restituisce uno dei seguenti `KMSInvalidStateException` 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 delle 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 APIspecifica del proxy dell'archivio chiavi AWS KMS esterno, o se non funziona correttamente o non è in grado di comunicare con esso AWS KMS, non sarà possibile creare o utilizzare KMS le chiavi nell'archivio 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 facilitare la diagnosi dei problemi relativi al proxy, AWS KMS fornisce XKSProxy Text Client, un client di test open source che verifica che il proxy dell'archivio chiavi esterno sia conforme alla specifica del proxy AWS KMS External Key Store. API

`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. È possibile trovare le voci di CloudTrail registro relative all' AWS KMS APIoperazione 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 GetHealthStatusrisposta) 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. Tuttavia, se persiste, potrebbe indicare che il proxy dell'archivio chiavi esterno non è conforme alla specifica del proxy dell'archivio chiavi AWS KMS esterno. API 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 del proxy API dell'archivio chiavi AWS KMS esterno. 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 `awsSourceVpce` campi `awsSourceVpc` and sono inclusi nei metadati solo quando la richiesta proviene da un VPC endpoint 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'operazione correlata fallisce. AWS KMS `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 te 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

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

Per impostazione predefinita, AWS KMS genera e protegge il materiale delle KMS chiavi crittografiche. 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 KMS chiave, per impostazione predefinita, AWS KMS genera e protegge il materiale crittografico relativo alla KMS chiave.

Per proteggere il materiale chiave per KMS le chiavi, AWS KMS si affida a una flotta distribuita di FIPS140-2 moduli di sicurezza hardware convalidati dal livello di sicurezza 3 (). HSMs Ciascuno AWS KMS HSM è un dispositivo hardware indipendente dedicato progettato per fornire funzioni crittografiche dedicate per soddisfare i requisiti di sicurezza e scalabilità di. AWS KMS(I dispositivi HSMs AWS KMS utilizzati nelle regioni cinesi sono certificati OSCCAe conformi a tutte le normative cinesi pertinenti, ma non sono convalidati nell'ambito del programma di convalida dei moduli crittografici FIPS 140-2.)

Il materiale chiave di una KMS chiave viene crittografato per impostazione predefinita quando viene generato in. HSM Il materiale chiave viene decrittografato solo all'interno della memoria HSM volatile 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 al suo interno HSM e trasferito su un sistema di archiviazione persistente a bassa latenza e altamente durevole (99,25%), dove rimane separato e isolato dal. HSMs Il materiale chiave in testo semplice non esce mai dai limiti di HSM sicurezza; non viene mai scritto su disco o memorizzato in 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 a chiave crittografica 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 in testo non crittografato contenente le

chiavi del cliente AWS KMS viene utilizzato per le operazioni crittografiche, all'interno del quale è AWS KMS FIPS stato 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 e non trasferibile proprietario 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 IAM politiche 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 KMS chiavi supportate da materiale chiave generato e utilizzato all'esterno di AWS KMS. KMSle AWS CloudHSM chiavi negli archivi delle chiavi sono supportate da chiavi nei moduli di sicurezza AWS CloudHSM hardware controllati dall'utente. Queste HSMs sono certificate al livello di sicurezza FIPS 140-2 3. KMSle chiavi 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 dispositivo fisico HSM nel tuo data center privato.

Un'altra funzionalità opzionale consente di importare il materiale chiave per una KMS chiave. 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 RSA chiavi generata in un AWS KMS HSM. Il materiale chiave importato viene decrittografato in un file AWS KMS HSM e ricrittografato con una chiave simmetrica in. 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 KMS le chiavi non esce mai dal limite del non criptato. AWS KMS HSMs Non viene esportato o trasmesso in nessuna operazione. AWS KMS API

L'eccezione è rappresentata dalle chiavi multiregionali, in cui viene AWS KMS utilizzato un meccanismo di replica interregionale per copiare il materiale chiave di una chiave multiregionale da una chiave HSM in una Regione AWS all'altra. HSM 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 AWS KMS keys di sicurezza hardware conformi al livello di sicurezza FIPS 140-2 (). HSMs L'unica eccezione è rappresentata dalle regioni cinesi, dove i HSMs dati AWS KMS utilizzati per generare KMS le chiavi sono conformi a tutte le normative cinesi pertinenti, ma non sono convalidati ai sensi del 140-2 Cryptographic Module Validation Program. FIPS Quando non viene utilizzato, il materiale chiave viene crittografato da una HSM chiave e scritto su un dispositivo di archiviazione durevole e persistente. Il materiale chiave per KMS le chiavi e le chiavi di crittografia che proteggono il materiale chiave non viene mai rilasciato HSMs in formato testo semplice.

La crittografia e la gestione del materiale chiave per KMS le chiavi sono gestite 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 KMS le chiavi non viene mai esportato o trasmesso durante le operazioni. AWS KMS API AWS KMS utilizza identificatori di chiave per rappresentare le KMS chiavi nelle API operazioni. Allo stesso modo, il materiale chiave per KMS le chiavi negli archivi di chiavi AWS KMS personalizzati non è esportabile e non viene mai trasmesso nelle AWS KMS nostre operazioni. AWS CloudHSM API

Tuttavia, alcune AWS KMS API operazioni restituiscono chiavi di dati. Inoltre, i clienti possono utilizzare API le operazioni per importare materiale chiave per KMS le chiavi selezionate.

Tutte le AWS KMS API chiamate 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 la tecnologia post-quantistica ibrida TLS per gli endpoint AWS KMS di servizio in tutte le regioni, ad eccezione delle regioni cinesi. AWS KMS non supporta la postquantistica TLS ibrida per gli endpoint in. FIPS 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 hai bisogno di FIPS 140-2 moduli crittografici convalidati per l'accesso AWS tramite un'interfaccia a riga di comando o un, usa un endpoint. API FIPS Per utilizzare endpoint o AWS KMS endpoint standard, i client AWS KMS FIPS devono supportare 1.2 o versioni successive. TLS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard () FIPS 140-2. Per un elenco degli AWS KMS FIPS endpoint, vedere AWS Key Management Service 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, consulta Sicurezza delle comunicazioni interne in Cryptographic Details. AWS Key Management Service

Riservatezza del traffico Internet

AWS KMS supporta una AWS Management Console serie di API operazioni 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 IPSec VPN connessione tramite Internet
- AWS Direct Connect, che collega la rete interna a un' AWS Direct Connect ubicazione tramite un cavo Ethernet standard in fibra ottica.

Tutte le AWS KMS API chiamate 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 KMS le chiavi è consentito solo da AWS KMS API host noti sulla rete AWS interna.

Per connetterti direttamente AWS KMS dal tuo cloud privato virtuale (VPC) senza inviare traffico sulla rete Internet pubblica, utilizza gli VPC endpoint, forniti da AWS PrivateLink. Per ulteriori informazioni, consulta Connect a AWS KMS tramite un VPC endpoint.

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 AWS KMS API endpoint.

Gestione delle identità e degli accessi per AWS Key Management Service

AWS Identity and Access Management (IAM) consente di controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (dispone delle autorizzazioni) a utilizzare le risorse. AWS KMS Per ulteriori informazioni, consulta Utilizzo IAM delle politiche con AWS KMS.

Le politiche chiave sono il meccanismo principale per controllare l'accesso alle KMS chiavi in ingresso. AWS KMS Ogni KMS chiave deve avere una politica chiave. Puoi anche utilizzare IAMpolitiche e sovvenzioni, insieme alle politiche chiave, per controllare l'accesso alle tue KMS chiavi. Per ulteriori informazioni, consulta KMSaccesso con chiavi e autorizzazioni.

Se utilizzi un Amazon Virtual Private Cloud (AmazonVPC), puoi creare un VPC endpoint di interfaccia AWS KMS su powered by AWS PrivateLink. Puoi anche utilizzare le policy VPC degli endpoint per determinare quali principali utenti possono accedere al tuo AWS KMS endpoint, quali API chiamate possono effettuare e a quale KMS chiave 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 Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi 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 API operazioni per i servizi esistenti.

Per ulteriori informazioni, consulta Policy gestite da AWSnellaGuida per l'utente IAM.

AWS politica gestita: AWSKeyManagementServicePowerUser

È possibile collegare la policy `AWSKeyManagementServicePowerUser` alle identità IAM.

Puoi utilizzare la policy `AWSKeyManagementServicePowerUser` gestita per concedere IAM ai responsabili del tuo account le autorizzazioni di un power user. Gli utenti esperti possono creare KMS chiavi, utilizzare e gestire le KMS chiavi che creano e visualizzare tutte le KMS chiavi e IAM le identità. I responsabili che dispongono della politica `AWSKeyManagementServicePowerUser` gestita possono anche ottenere autorizzazioni da altre fonti, tra cui politiche chiave, altre IAM politiche e sovvenzioni.

`AWSKeyManagementServicePowerUser`è una politica gestita AWS . IAM Per ulteriori informazioni sulle politiche AWS gestite, vedere le politiche AWS gestite nella Guida IAM per l'utente.

> **Note**
>
> Le autorizzazioni contenute in questa policy, che sono specifiche per una KMS chiave, ad esempio `kms:TagResource` e`kms:GetKeyRotationStatus`, sono efficaci solo quando la policy chiave per quella KMS chiave consente esplicitamente di utilizzare IAM le policy per controllare l'accesso alla chiave. Account AWS Per determinare se un'autorizzazione è specifica per una KMS chiave, consulta AWS KMS autorizzazioni e cerca un valore di KMSchiave nella colonna Risorse.
>
> Questa politica fornisce a un power user le autorizzazioni per qualsiasi KMS chiave con una politica chiave che ne consente l'operazione. Per le autorizzazioni tra più account, come `kms:DescribeKey` e`kms:ListGrants`, ciò potrebbe includere KMS le chiavi in untrusted. Account AWS Per informazioni dettagliate, consulta Best practice per le policy IAM e Consentire agli utenti di altri account di utilizzare una KMS chiave. Per determinare se un'autorizzazione è valida per KMS le chiavi di altri account, consulta AWS KMS autorizzazioni e cerca il valore Sì nella colonna Utilizzo tra account.
>
> Per consentire ai responsabili di visualizzare la AWS KMS console senza errori, l'amministratore deve utilizzare il tag: GetResources permission, che non è incluso

nella `AWSKeyManagementServicePowerUser` policy. È possibile consentire questa autorizzazione in una IAM politica separata.

La IAM politica AWSKeyManagementServicePowerUsergestita include le seguenti autorizzazioni.

- Consente ai principali di creare KMS chiavi. Poiché questo processo include l'impostazione della politica delle chiavi, gli utenti esperti possono concedere a se stessi e agli altri il permesso di utilizzare e gestire le KMS chiavi che creano.
- Consente ai responsabili di creare ed eliminare alias e tag su tutte le KMS chiavi. La modifica di un tag o di un alias può consentire o negare l'autorizzazione all'uso e alla gestione della chiave. KMS Per informazioni dettagliate, consultare ABACper AWS KMS.
- Consente ai responsabili di ottenere informazioni dettagliate su tutte le KMS chiavi, inclusa la chiave, la configurazione crittograficaARN, la politica delle chiavi, gli alias, i tag e lo stato di rotazione.
- Consente ai responsabili di elencare IAM utenti, gruppi e ruoli.
- Questa politica non consente ai mandanti di utilizzare o gestire KMS chiavi che non hanno creato. Tuttavia, possono modificare alias e tag su tutte le KMS chiavi, il che potrebbe consentire o negare loro l'autorizzazione a utilizzare o gestire una chiave. KMS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateAlias",
                "kms:CreateKey",
                "kms:DeleteAlias",
                "kms:Describe*",
                "kms:GenerateRandom",
                "kms:Get*",
                "kms:List*",
                "kms:TagResource",
                "kms:UntagResource",
                "iam:ListGroups",
                "iam:ListRoles",
                "iam:ListUsers"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politica gestita: AWSServiceRoleForKeyManagementServiceCustomKeyStores

Non puoi collegare `AWSServiceRoleForKeyManagementServiceCustomKeyStores` alle tue 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 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: AWSServiceRoleForKeyManagementServiceMultiRegionKeys

Non puoi collegare `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` alle tue 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 ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al RSS feed sulla AWS KMS Cronologia dei documenti pagina.

Modifica	Descrizione	Data
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy: aggiornamento a policy esistente	AWS KMS ha aggiunto un campo statement ID (`Sid`) alla politica gestita nella versione della policy v2.	21 novembre 2024
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy: aggiornamento a policy esistente	AWS KMS ha aggiunto `ec2:DescribeVpcs ec2:DescribeNetworkAcls` , e	10 novembre 2023

Modifica	Descrizione	Data
	`ec2:DescribeNetworkInterfaces` le autorizzazioni per monitorare le modifiche nel file VPC che contiene il AWS CloudHSM cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di errori.	
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 al servizio è un tipo unico di IAM ruolo 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 conto dell'utente.

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 trust e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo avere 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 su altri servizi che supportano i ruoli collegati ai servizi, vedi AWS Servizi compatibili con IAM e cerca i servizi con Sì nella colonna Ruolo collegato 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

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 AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato al servizio nel tuo. Account AWS Gli utenti che creano archivi di AWS CloudHSM chiavi devono disporre dell'`iam:CreateServiceLinkedRole`autorizzazione che consenta loro di creare ruoli collegati ai servizi.

Per visualizzare i dettagli sugli aggiornamenti della politica AWSKeyManagementServiceCustomKeyStoresServiceRolePolicygestita, vedere. AWS KMS aggiornamenti alle politiche AWS gestite

Argomenti

Informazioni sul ruolo AWS KMS collegato al servizio

Un ruolo collegato al servizio è un IAM ruolo che concede a un AWS servizio l'autorizzazione a chiamare altri AWS servizi per conto dell'utente. È progettato per semplificare l'utilizzo delle funzionalità di più AWS servizi integrati senza dover creare e mantenere politiche complesseIAM. 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 connetti un archivio di AWS CloudHSM chiavi per consentire l'accesso alla rete dall' AWS KMS interno del cluster VPC che contiene il AWS CloudHSM cluster.
- ec2: CreateNetworkInterface — utilizzato quando si connette un archivio di AWS CloudHSM chiavi per creare l'interfaccia di rete utilizzata per la comunicazione tra AWS KMS e il AWS CloudHSM cluster.
- 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 gruppo di sicurezza VPC che contiene il AWS CloudHSM cluster in modo che AWS KMS possa fornire messaggi di errore chiari in caso di guasti.
- ec2: DescribeVpcs — utilizzato per monitorare le modifiche nel cluster VPC che contiene il AWS CloudHSM 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 VPC che contiene il AWS CloudHSM cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di guasti.
- ec2: DescribeNetworkInterfaces — utilizzato per monitorare i cambiamenti nelle interfacce di rete AWS KMS create nel cluster VPC che contiene il AWS CloudHSM cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di guasti.

```
{
  "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 AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato al servizio è affidabile`ecks.kms.amazonaws.com`, solo AWS KMS può assumere questo ruolo collegato al servizio. 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 AWSServiceRoleForKeyManagementServiceCustomKeyStoresruolo 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, vedere Utilizzo dei ruoli collegati ai servizi nella Guida per l'utente. IAM

Creazione del ruolo collegato ai servizi

AWS KMS crea automaticamente il ruolo AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato 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 al servizio nella Guida per l'utente. IAM

Eliminazione del ruolo collegato ai servizi

AWS KMS non elimina il ruolo AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato 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 AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato 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 AWSKeyManagementServiceMultiRegionKeysServiceRolePolicygestita, 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 al servizio è un IAM ruolo 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 complesseIAM.

Per le chiavi multiregionali, AWS KMS crea il ruolo AWSServiceRoleForKeyManagementServiceMultiRegionKeyscollegato ai servizi con la AWSKeyManagementServiceMultiRegionKeysServiceRolePolicypolicy 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 AWSServiceRoleForKeyManagementServiceMultiRegionKeyscollegato al servizio è affidabile`mrk.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 KMS chiave.

Per ulteriori informazioni su come AWS i servizi utilizzano i ruoli collegati ai servizi, vedere Using Service-Linked Roles nella Guida per l'utente. IAM

```
{
    "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 AWSServiceRoleForKeyManagementServiceMultiRegionKeyscollegato ai servizi 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 AWSServiceRoleForKeyManagementServiceMultiRegionKeyscollegato 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. IAM

Eliminazione del ruolo collegato ai servizi

AWS KMS non elimina il ruolo AWSServiceRoleForKeyManagementServiceMultiRegionKeyscollegato al servizio dal tuo Account AWS e non è possibile eliminarlo. Tuttavia, AWS KMS non assume il AWSServiceRoleForKeyManagementServiceMultiRegionKeysruolo né utilizza alcuna delle sue autorizzazioni a meno che non si disponga di chiavi multiregionali nella propria regione. Account AWS

Registrazione e monitoraggio in 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 soluzioni AWS. AWS fornisce diversi strumenti per monitorare le chiavi KMS.

File di log di AWS CloudTrail

> Ogni chiamata a un'operazione dell'API AWS KMS viene acquisita come evento in un log di AWS CloudTrail. Questi log registrano tutte le chiamate API dalla console AWS KMS e le chiamate effettuate da AWS KMS e altri servizi AWS. Le chiamate API tra account, ad esempio una chiamata per utilizzare una chiave KMS in un'altraAccount AWS, vengono registrate nei CloudTrail log 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 file di log da AWS CloudTrail e altre origini. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

InfattiAWS 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 i tasti con CloudWatch”.

Amazon EventBridge

AWS KMSgenera EventBridge eventi quando la chiave KMS viene ruotata o eliminata o il materiale chiave importato nella chiave KMS scade. Cerca gli eventi AWS KMS (operazioni API) e instradali a una o più funzioni o flussi di destinazione per acquisire le 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

Puoi monitorare l'uso di AWS KMS per verificare gli standard del settore della sicurezza e la conformità alle procedure consigliate utilizzando AWS Security Hub. Security Hub utilizza 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.

Convalida della conformità per AWS Key Management Service

Revisori di terze parti valutano la sicurezza e la conformità di AWS Key Management Service come parte di più programmi di conformità di AWS. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Argomenti

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 informazioni sull'utilizzo di AWS Artifact, consulta Download di report in AWS Artifact.

Ulteriori informazioni

La tua responsabilità di conformità durante l'utilizzo di AWS KMS è determinata dalla riservatezza dei dati, dagli obiettivi dell'azienda e dalle leggi e normative applicabili. Se l'utilizzo di AWS KMS è soggetto a conformità con uno standard pubblicato, AWS fornisce risorse utili:

- Servizi AWS coperti dal programma di conformità – Questa pagina elenca i servizi AWS che rientrano nell'ambito di programmi di conformità specifici. Per informazioni generali, consulta Programmi di conformità di AWS.
- Guide Quick Start per la sicurezza e conformità: queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono fasi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Risorse per la conformità di AWS: questa raccolta di workbook e guide potrebbe essere utile al settore e alla posizione.
- AWS Config: questo servizio AWS valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti di settore.
- AWS Security Hub: questo servizio AWS fornisce una vista completa dello stato di sicurezza in AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWSe verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.

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. KMS APIs Per adempiere a questo impegno, AWS KMS garantisce che tutti i dati e le informazioni di autorizzazione necessari per eseguire una API richiesta 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 KMS chiave vengono replicate su tutti gli host della regione per garantire che la richiesta successiva possa essere elaborata correttamente da qualsiasi host della regione. Le richieste di operazioni crittografiche che utilizzano la KMS chiave 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 la disponibilità del 99,999% e di sostenere tassi di richieste elevatiSLA, proteggendo al contempo la riservatezza delle chiavi e dei dati.

Vengono implementati diversi meccanismi di rafforzamento dell'integrità per garantire che la KMS chiave specificata per l'operazione di crittografia sia sempre quella utilizzata.

Il materiale chiave in testo non crittografato per le chiavi è ampiamente protetto. KMS Il materiale chiave viene crittografato non HSM appena viene creato e il materiale chiave crittografato viene immediatamente spostato in uno spazio di archiviazione sicuro a bassa latenza. La chiave crittografata viene recuperata e decrittografata nel momento HSM giusto per l'uso. La chiave di testo in chiaro rimane in HSM memoria solo per il tempo necessario a completare l'operazione di crittografia. Quindi viene nuovamente crittografata in HSM e la chiave crittografata viene restituita all'archivio. Il materiale chiave in chiaro non esce mai dal fileHSMs; non viene mai scritto nell'archivio persistente.

Le migliori pratiche di resilienza in AWS KMS

Per ottimizzare la resilienza AWS KMS delle risorse, prendi in considerazione le seguenti strategie.

- Per supportare la tua strategia di backup e disaster recovery, prendi in considerazione le chiavi multiregionali, che sono KMS chiavi create in un'unica area Regione AWS e replicate solo nelle regioni specificate. Con le chiavi multiregionali, è possibile spostare risorse crittografate tra Regioni AWS (all'interno della stessa partizione) senza mai esporre il testo non crittografato 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, incluse politiche e politiche chiave. IAM Inoltre, puoi inviare le tue richieste AWS KMS utilizzando un'VPCinterfaccia endpoint fornita da. AWS PrivateLink Quando lo fai, tutte le comunicazioni tra Amazon VPC e Amazon AWS KMS vengono condotte interamente all'interno della AWS rete utilizzando un AWS KMS endpoint dedicato limitato al tuoVPC. Puoi proteggere ulteriormente queste richieste creando un livello di autorizzazione aggiuntivo utilizzando le policy VPCdegli endpoint. Per i dettagli, consulta Connessione AWS KMS tramite un VPC endpoint.

Sicurezza dell'infrastruttura nell'AWS Key Management Service

In qualità di servizio gestito, AWS Key Management Service (AWS KMS) è protetto dalle procedure di sicurezza di rete globali di AWS descritte nel whitepaper Panoramica delle procedure di sicurezza di Amazon Web Services.

Per accedere a AWS KMS tramite la rete, puoi richiamare le operazioni dell'API AWS KMS descritte nella Documentazione di riferimento dell'API AWS Key Management Service. AWS KMS richiede l'utilizzo di TLS 1.2 e suggerisce l'utilizzo di TLS 1.3 in tutte le regioni. AWS KMS supporta anche TLS ibrido post-quantistico per gli endpoint del servizio AWS KMS in tutte le regioni, ad eccezione delle regioni cinesi. AWS KMS non supporta il protocollo TLS post-quantistico ibrido 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. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi chiamare queste operazioni API da qualsiasi percorso di rete, ma AWS KMS supporta le condizioni delle 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 impedire ad AWS l'utilizzo della chiave KMS per tuo conto. Per informazioni dettagliate, vedi AWS chiavi di condizione globali.

Ad esempio, l'istruzione della policy chiave seguente consente agli utenti che possono assumere il ruolo `KMSTestRole` di utilizzare questa AWS KMS key per le operazioni di crittografia specificate, a meno che l'indirizzo IP di origine non sia uno degli indirizzi IP specificati nella policy.

```
{
  "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 utilizzata da AWS KMS è soggetta ai controlli descritti nella sezione Sicurezza fisica e ambientale del whitepaper Panoramica delle procedure di sicurezza di Amazon Web Services. Puoi trovare altri dettagli nei report di conformità e nei risultati degli audit di terze parti elencati nella sezione precedente.

AWS KMS è supportato dai moduli HSM (Hardened Hardware Security Module) dedicati progettati con controlli specifici per resistere agli attacchi fisici. I moduli HSM sono dispositivi fisici che non dispongono di un livello di virtualizzazione, ad esempio un hypervisor, che condivide il dispositivo fisico tra diversi tenant logici. Il materiale chiave per le AWS KMS keys viene archiviato solo nella memoria volatile sui moduli HSM e solo quando la 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 sull'operazione dei moduli HSM AWS KMS, consulta i Dettagli crittografici di AWS Key Management Service..

Quote

Per renderla AWS KMS reattiva 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, AWS Support visita il Centro e crea un caso.

Argomenti

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, AWS Support 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	50	Alias creati dal cliente	Sì
Sovvenzioni 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	10	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, consultare 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 KMS chiave è considerata una risorsa. Chiavi gestite da AWSe Chiavi di proprietà di AWSnon rientrano in questa quota.

Alias per KMS chiave: 50

Puoi associare fino a 50 alias a ciascuna chiave gestita dal cliente. Gli alias AWS associati a Chiavi gestite da AWSnon 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 KMS chiave è conforme a questa quota. Se una KMS chiave supera questa quota, ai mandanti autorizzati a utilizzare

la KMS chiave in base alla `kms:ResourceAliases` condizione viene negato l'accesso alla chiave. KMS Per informazioni dettagliate, consultare Accesso negato a causa di quota alias.

La quota Alias per KMS chiave 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 di alias per regione.

Sovvenzioni per KMS chiave: 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 autorizzate da sovvenzioni che utilizzano la stessa KMS chiave contemporaneamente. Una volta raggiunta la quota, è possibile creare nuove sovvenzioni sulla KMS chiave solo quando una sovvenzione attiva viene ritirata o revocata.

Ad esempio, quando colleghi un volume Amazon Elastic Block Store (AmazonEBS) a un'istanza Amazon Elastic Compute Cloud (AmazonEC2), il volume viene decrittografato in modo da poterlo leggere. Per ottenere l'autorizzazione a decrittografare i dati, Amazon EBS crea una concessione per ogni volume. Pertanto, se tutti i tuoi EBS volumi Amazon utilizzano la stessa KMS chiave, non puoi allegare più di 50.000 volumi contemporaneamente.

Quote di risorse per gli archivi delle chiavi personalizzate: 10

È possibile creare fino a 10 archivi di chiavi personalizzati in ciascuna Account AWS regione. Se si tenta di crearne altri, l'CreateCustomKeyStoreoperazione 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 dei tasti su richiesta un massimo di 10 volte per KMS chiave. Se si tenta di eseguire più rotazioni su richiesta, l'RotateKeyOnDemandoperazione non riesce.

Questa quota non è regolabile. Non è possibile aumentarlo utilizzando Service Quotas o creando un case in. AWS Support 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 API operazioni richieste al secondo. Le quote di richiesta variano in base all'APIoperazione, al Regione AWS e ad altri fattori, come il KMS tipo di chiave. Quando si supera una quota di API richiesta, AWS KMS limita la 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, AWS Support visita il Centro e crea un caso.

Se stai superando la quota richiesta per l'GenerateDataKeyoperazione, 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 Gestire le tariffe delle AWS KMS API richieste utilizzando Service Quotas e Amazon CloudWatch nel blog sulla AWS sicurezza.

Argomenti

Richiedi quote per ogni operazione AWS KMS API

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 KMS chiave 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`	1.000 (condivisi) per le chiavi RSA KMS

Nome quota	Valore predefinito (richieste al secondo)
Si applica a: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	
`Cryptographic operations (ECC and SM2) request rate` Si applica a: • `Decrypt`—supportato solo per le chiavi SM2 (solo per le regioni cinesi) KMS • `DeriveSharedSecret` • `Encrypt`—supportato solo per le chiavi SM2 (solo per le regioni cinesi) KMS • `ReEncrypt` —supportato solo per le chiavi SM2 (solo per le regioni cinesi) KMS • `Sign` • `Verify`	1.000 (condivise) per le chiavi a curva ellittica (ECC) e SM2 (solo regioni cinesi) KMS

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)
`GenerateDataKeyPair (ECC_SECG_P256K1) request rate` Si applica a: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (RSA_2048) request rate` Si applica a: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	1
`GenerateDataKeyPair (RSA_3072) request rate` Si applica a: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,5 (1 in ogni intervallo di 2 secondi)
`GenerateDataKeyPair (RSA_4096) request rate` Si applica a: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,1 (1 in ogni intervallo di 10 secondi)

Nome quota	Valore predefinito (richieste al secondo)
`GenerateDataKeyPair (SM2 – China Regions only) request rate` Si applica a: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	25
`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 AWSnon rientra nelle quote richieste per l'utente Account AWS, anche quando vengono utilizzate per proteggere le risorse dell'account.
- Le quote di richiesta si applicano alle richieste inviate a FIPS endpoint e non endpoint. FIPS Per un elenco degli endpoint del AWS KMS servizio, consulta AWS Key Management Service endpoint e quote in. Riferimenti generali di AWS
- La limitazione si basa su tutte le richieste relative a KMS chiavi di tutti i tipi nella regione. Questo totale include le richieste provenienti da tutti i principali attori della regione Account AWS, comprese le richieste provenienti dai AWS servizi per conto dell'utente.
- Ogni quota di richieste è calcolata in modo indipendente. Ad esempio, le richieste per l'CreateKeyoperazione non hanno alcun effetto sulla quota di richieste per l'CreateAliasoperazione. 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 crittografiche su KMS chiavi simmetriche più 5 `EnableKey` operazioni al secondo senza subire limitazioni.

Quote condivise per le operazioni di crittografia

AWS KMS le operazioni crittografiche condividono le quote di richiesta. È possibile richiedere qualsiasi combinazione delle operazioni crittografiche supportate dalla KMS chiave, in modo che il numero totale di operazioni crittografiche non superi la quota di richiesta per quel tipo di chiave. KMS Le eccezioni sono GenerateDataKeyPaire GenerateDataKeyPairWithoutPlaintext, che condividono una quota separata.

Le quote per diversi tipi di KMS chiavi vengono calcolate indipendentemente. 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.

- La frequenza di richiesta delle operazioni crittografiche (simmetriche) è la quota di richieste condivise per le operazioni crittografiche che utilizzano chiavi KMS simmetriche in un account e in una regione. Questa quota si applica alle operazioni crittografiche con chiavi e chiavi di crittografia simmetriche, anch'esse simmetriche. HMAC

Ad esempio, è possibile utilizzare KMSchiavi simmetriche in un file Regione AWS con una quota condivisa di 10.000 richieste al secondo. Quando si effettuano 7.000 GenerateDataKeyrichieste al secondo e 2.000 richieste di decriptazione al secondo, AWS KMS non limita le 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 crittografiche sulle chiavi di crittografia simmetriche in un archivio di KMS chiavipersonalizzato vengono conteggiate sia per il tasso di richieste di operazioni crittografiche (simmetriche) per l'account sia per la quota di richieste di archiviazione delle chiavi personalizzate per l'archivio di chiavi personalizzato.

- La frequenza di richiesta delle operazioni crittografiche (RSA) è la quota di richieste condivisa per le operazioni crittografiche che utilizzano chiavi asimmetriche. RSA KMS

  Ad esempio, con una quota di richieste di 1.000 operazioni al secondo, è possibile effettuare 400 richieste di crittografia e 200 richieste di decrittografia con RSA KMS chiavi in grado di crittografaree decrittografare, oltre a 250 richieste di firma e 150 richieste di verifica con chiavi in grado di firmare e verificare. RSA KMS
- La frequenza di richiesta delle operazioni crittografiche (ECC) è la quota di richieste condivisa per le operazioni crittografiche che utilizzano chiavi asimmetriche a curva ellittica () e chiavi asimmetriche SM. ECC KMS KMS

  Ad esempio, con una quota di richieste di 1.000 operazioni al secondo, puoi effettuare 400 richieste Sign e 200 richieste Verify con ECC KMS chiavi in grado di firmare e verificare, oltre a 250 richieste Sign e 150 richieste Verify con chiavi in grado di firmare e verificare. SM2 KMS
- La quota di richiesta di archiviazione chiavi personalizzata è la quota di richiesta condivisa per le operazioni crittografiche sulle KMS chiavi in un archivio di chiavi personalizzato. Questa quota viene calcolata separatamente per ogni archivio delle chiavi personalizzate.

  Le operazioni crittografiche sulle chiavi di crittografia simmetriche in un archivio KMS chiavi personalizzato vengono conteggiate sia per la frequenza di richieste di operazioni crittografiche (simmetriche) per l'account che per la quota di richieste di archiviazione chiavi personalizzate per l'archivio chiavi personalizzato.

Anche le quote per i diversi tipi di chiave sono calcolate in modo indipendente. Ad esempio, nella regione Asia Pacifico (Singapore), se utilizzi KMS chiavi simmetriche e asimmetriche, puoi effettuare fino a 10.000 chiamate al secondo con chiavi simmetriche (KMSHMACchiavi incluse) più fino a 500

chiamate aggiuntive al secondo con le tue KMS chiavi RSA asimmetriche, più fino a 300 richieste aggiuntive al secondo con le tue KMS chiavi ECC basate.

APIrichieste effettuate per tuo conto

Puoi effettuare API richieste direttamente o utilizzando un AWS servizio integrato che effettua API richieste per tuo AWS KMS conto. La quota si applica a entrambi i tipi di richieste.

Ad esempio, potresti archiviare dati in Amazon S3 utilizzando la crittografia lato server con una KMS chiave (-). SSE KMS Ogni volta che carichi o scarichi un oggetto S3 crittografato con SSE -KMS, Amazon S3 invia `GenerateDataKey` una richiesta (per i caricamenti) `Decrypt` o (per i download) AWS KMS a tuo nome. Queste richieste vengono conteggiate ai fini della tua quota, AWS KMS quindi 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 KMS chiave di proprietà di un altro account, si parla di richiesta tra account. Per le richieste tra più account, AWS KMS limita l'account che effettua le richieste, non l'account che possiede la chiave. KMS Ad esempio, se un'applicazione nell'account A utilizza una KMS chiave nell'account B, l'uso della KMS chiave si applica solo alle quote nell'account A.

Quote di richiesta per l'archivio delle chiavi personalizzate

AWS KMS mantiene le quote di richiesta per le operazioni crittografiche sulle KMS chiavi 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 è possibile visualizzare o gestire queste quote utilizzando le operazioni Service API Quotas. Per richiedere una modifica delle quote di richiesta per l'archivio delle chiavi esterne, visita il Centro AWS Support e crea un caso.
> Se il AWS CloudHSM cluster associato a un archivio AWS CloudHSM chiavi elabora 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 operazione crittografica su una KMS chiave in un archivio di chiavi personalizzato viene conteggiata ai fini di due quote:

- Quota di frequenza della richiesta (per account) di operazioni di crittografia (simmetriche)

  Le richieste di operazioni crittografiche sulle KMS chiavi in un archivio di chiavi personalizzato vengono conteggiate ai fini della `Cryptographic operations (symmetric) request rate` quota per ciascuna Account AWS regione. Ad esempio, negli Stati Uniti orientali (Virginia settentrionale) (us-east-1), Account AWS ognuno può ricevere fino a 50.000 richieste al secondo su chiavi di KMS crittografia simmetriche, incluse le richieste che utilizzano KMS una chiave in un archivio di chiavi personalizzato.

- Quota di richiesta dell'archivio di chiavi personalizzate (per archivio delle chiavi personalizzate)

  Le richieste di operazioni crittografiche sulle KMS chiavi in un archivio di chiavi personalizzato contano anche per un totale di 1.800 operazioni al secondo. `Custom key store request quota` Tali quote vengono calcolate separatamente per ogni archivio delle chiavi personalizzate. Potrebbero includere richieste provenienti da più utenti Account AWS che utilizzano KMS chiavi nell'archivio chiavi personalizzato.

Ad esempio, un'operazione di crittografia su una KMS chiave in un archivio di 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 (50.000 richieste al secondo) per l'account e la regione e per una `Custom key store request quota` (1.800 richieste al secondo) per il relativo archivio chiavi personalizzato. Tuttavia, una richiesta per un'operazione di gestione PutKeyPolicy, ad esempio su una KMS chiave 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 AWS KMS risposte rapide e affidabili alle API richieste di tutti i clienti, limita le API richieste 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 quello 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. Questa strategia viene implementata automaticamente per HTTP 400 errori in alcuni casi. AWS SDKs
- Una frequenza ininterrotta o sostenuta di richieste di modifica dello stato della stessa KMS chiave. Questa condizione è spesso nota come "tasto di scelta rapida".

  Ad esempio, se un'applicazione del tuo account invia una raffica persistente di `EnableKey` `DisableKey` richieste per la stessa KMS chiave, limita le richieste AWS KMS . 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 richieste o consolidi le richieste di più funzioni.

- Le richieste di operazioni sulle KMS chiavi 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 KMS chiavi in un AWS CloudHSM archivio di chiavi quando non ci sono 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 Gestire le tariffe delle AWS KMS API richieste 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 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, AWS Support 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 è possibile visualizzare o gestire queste quote utilizzando le operazioni Service API Quotas. Per richiedere una modifica delle quote di richiesta per l'archivio delle chiavi esterne, visita il Centro AWS Support e crea un caso.

Esempi di codice per AWS KMS l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare 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. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Nozioni di base

Salve AWS Key Management Service

L'esempio di codice seguente mostra come iniziare a utilizzare AWS Key Management Service.

Java

SDKper 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.

```
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 API i dettagli, vedi ListKeysin AWS SDK for Java 2.x APIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi ListKeysin AWS SDK for PHP APIReference.

Esempi di codice

- Esempi di base per AWS KMS l'utilizzo AWS SDKs
  - Salve AWS Key Management Service
  - Impara le nozioni di base di AWS KMS con un AWS SDK
  - Azioni per AWS KMS l'utilizzo AWS SDKs
    - Utilizzare CreateAlias con un AWS SDK o CLI
    - Utilizzare CreateGrant con un AWS SDK o CLI
    - Utilizzare CreateKey con un AWS SDK o CLI
    - Utilizzare Decrypt con un AWS SDK o CLI
    - Utilizzare DeleteAlias con un AWS SDK o CLI
    - Utilizzare DescribeKey con un AWS SDK o CLI
    - Utilizzare DisableKey con un AWS SDK o CLI
    - Utilizzare EnableKey con un AWS SDK o CLI
    - Utilizzare EnableKeyRotation con un AWS SDK o CLI
    - Utilizzare Encrypt con un AWS SDK o CLI
    - Utilizzare GenerateDataKey con un AWS SDK o CLI
    - Utilizzare GenerateDataKeyWithoutPlaintext con un AWS SDK o CLI
    - Utilizzare GenerateRandom con un AWS SDK o CLI
    - Utilizzare GetKeyPolicy con un AWS SDK o CLI
    - Utilizzare ListAliases con un AWS SDK o CLI
    - Utilizzare ListGrants con un AWS SDK o CLI
    - Utilizzare ListKeyPolicies con un AWS SDK o CLI
    - Utilizzare ListKeys con un AWS SDK o CLI
    - Utilizzare PutKeyPolicy con un AWS SDK o CLI
    - Utilizzare ReEncrypt con un AWS SDK o CLI
    - Utilizzare RetireGrant con un AWS SDK o CLI

- Utilizzare RevokeGrant con un AWS SDK o CLI
- Utilizzare ScheduleKeyDeletion con un AWS SDK o CLI
- Utilizzare Sign con un AWS SDK o CLI
- Utilizzare TagResource con un AWS SDK o CLI
- Utilizzare UpdateAlias con un AWS SDK o CLI
- Utilizzare Verify con un AWS SDK o CLI

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

- Salve AWS Key Management Service
- Impara le nozioni di base di AWS KMS con un AWS SDK
- Azioni per AWS KMS l'utilizzo AWS SDKs
  - Utilizzare CreateAlias con un AWS SDK o CLI
  - Utilizzare CreateGrant con un AWS SDK o CLI
  - Utilizzare CreateKey con un AWS SDK o CLI
  - Utilizzare Decrypt con un AWS SDK o CLI
  - Utilizzare DeleteAlias con un AWS SDK o CLI
  - Utilizzare DescribeKey con un AWS SDK o CLI
  - Utilizzare DisableKey con un AWS SDK o CLI
  - Utilizzare EnableKey con un AWS SDK o CLI
  - Utilizzare EnableKeyRotation con un AWS SDK o CLI
  - Utilizzare Encrypt con un AWS SDK o CLI
  - Utilizzare GenerateDataKey con un AWS SDK o CLI
  - Utilizzare GenerateDataKeyWithoutPlaintext con un AWS SDK o CLI
  - Utilizzare GenerateRandom con un AWS SDK o CLI
  - Utilizzare GetKeyPolicy con un AWS SDK o CLI
  - Utilizzare ListAliases con un AWS SDK o CLI

- Utilizzare ListGrants con un AWS SDK o CLI
- Utilizzare ListKeyPolicies con un AWS SDK o CLI
- Utilizzare ListKeys con un AWS SDK o CLI
- Utilizzare PutKeyPolicy con un AWS SDK o CLI
- Utilizzare ReEncrypt con un AWS SDK o CLI
- Utilizzare RetireGrant con un AWS SDK o CLI
- Utilizzare RevokeGrant con un AWS SDK o CLI
- Utilizzare ScheduleKeyDeletion con un AWS SDK o CLI
- Utilizzare Sign con un AWS SDK o CLI
- Utilizzare TagResource con un AWS SDK o CLI
- Utilizzare UpdateAlias con un AWS SDK o CLI
- Utilizzare Verify con un AWS SDK o CLI

Salve AWS Key Management Service

L'esempio di codice seguente mostra come iniziare a utilizzare AWS Key Management Service.

Java

SDKper 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.

```
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 API i dettagli, vedi ListKeysin AWS SDK for Java 2.x APIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi ListKeysin AWS SDK for PHP APIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Impara le nozioni di base di AWS KMS con un AWS SDK

Gli esempi di codice seguenti mostrano come:

- Crea una chiave KMS.
- Elenca KMS le chiavi del tuo account e ottieni dettagli su di esse.
- Abilita e disabilita KMS le chiavi.
- 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 KMS le chiavi.

Java

SDKper 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.

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::0000000000: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)
```

```
.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .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();
                            });
                    });
            })
            .exceptionally(throwable -> {
               throw new RuntimeException("Failed to sign or verify data",
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 API i dettagli, consultate i seguenti argomenti in AWS SDK for Java 2.x API Riferimento.
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Decrypt
  - DescribeKey
  - DisableKey
  - EnableKey
  - Encrypt
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Sign
  - TagResource

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.

```
        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 $caught){
            echo "There was a problem disabling the key: {$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 API i dettagli, consulta i seguenti argomenti in AWS SDK for PHP APIRiferimento.

- CreateAlias
- CreateGrant
- CreateKey
- Decrypt
- DescribeKey
- DisableKey
- EnableKey
- Encrypt
- GetKeyPolicy
- ListAliases
- ListGrants
- ListKeys
- RevokeGrant
- ScheduleKeyDeletion
- Sign
- TagResource

Python

SDKper Python (Boto3)

> **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.

```
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::0000000000: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 Wrapper e metodi per la gestione delle KMS chiavi.

```
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) -> str:
        """
        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: str) -> bytes:
        """
        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"
            ]
        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 API i dettagli, consulta i seguenti argomenti in AWS SDKPython (Boto3) Reference. API
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Decrypt
  - DescribeKey
  - DisableKey
  - EnableKey
  - Encrypt
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Sign
  - TagResource

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Azioni per AWS KMS l'utilizzo AWS 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.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta il AWS Key Management Service APIReference.

Esempi

- Utilizzare CreateAlias con un AWS SDK o CLI
- Utilizzare CreateGrant con un AWS SDK o CLI
- Utilizzare CreateKey con un AWS SDK o CLI
- Utilizzare Decrypt con un AWS SDK o CLI
- Utilizzare DeleteAlias con un AWS SDK o CLI
- Utilizzare DescribeKey con un AWS SDK o CLI
- Utilizzare DisableKey con un AWS SDK o CLI
- Utilizzare EnableKey con un AWS SDK o CLI
- Utilizzare EnableKeyRotation con un AWS SDK o CLI
- Utilizzare Encrypt con un AWS SDK o CLI
- Utilizzare GenerateDataKey con un AWS SDK o CLI
- Utilizzare GenerateDataKeyWithoutPlaintext con un AWS SDK o CLI
- Utilizzare GenerateRandom con un AWS SDK o CLI
- Utilizzare GetKeyPolicy con un AWS SDK o CLI
- Utilizzare ListAliases con un AWS SDK o CLI
- Utilizzare ListGrants con un AWS SDK o CLI
- Utilizzare ListKeyPolicies con un AWS SDK o CLI
- Utilizzare ListKeys con un AWS SDK o CLI

- Utilizzare PutKeyPolicy con un AWS SDK o CLI
- Utilizzare ReEncrypt con un AWS SDK o CLI
- Utilizzare RetireGrant con un AWS SDK o CLI
- Utilizzare RevokeGrant con un AWS SDK o CLI
- Utilizzare ScheduleKeyDeletion con un AWS SDK o CLI
- Utilizzare Sign con un AWS SDK o CLI
- Utilizzare TagResource con un AWS SDK o CLI
- Utilizzare UpdateAlias con un AWS SDK o CLI
- Utilizzare Verify con un AWS SDK o CLI

Utilizzare **`CreateAlias`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi CreateAliasin AWS SDK for .NET APIReference.

CLI

AWS CLI

Per creare un alias per una chiave KMS

Il `create-alias` comando seguente crea un alias denominato `example-alias` per la KMS chiave identificata dall'ID della chiave. `1234abcd-12ab-34cd-56ef-1234567890ab`

I nomi alias devono 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 alcun output. Per vedere il nuovo alias, usa il `list-aliases` comando.

Per ulteriori informazioni, vedere Using alias nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere CreateAliasin AWS CLI Command Reference.

Java

SDKper 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.

```
   /**
    * 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 API i dettagli, vedi CreateAliasin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun createCustomAlias(
    targetKeyIdVal: String?,
    aliasNameVal: String?,
) {
    val request =
```

```
        CreateAliasRequest {
            aliasName = aliasNameVal
            targetKeyId = targetKeyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        kmsClient.createAlias(request)
        println("$aliasNameVal was successfully created")
    }
}
```

- Per API i dettagli, vedi il riferimento CreateAlias AWSSDKa Kotlin API.

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 $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 API i dettagli, vedi CreateAliasin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere CreateAliasPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`CreateGrant`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi CreateGrantin AWS SDK for .NET APIReference.

CLI

AWS CLI

Per creare una sovvenzione

L'`create-grant`esempio seguente crea una concessione che consente all'`exampleUser`utente di utilizzare il `decrypt` comando sulla KMS chiave di `1234abcd-12ab-34cd-56ef-1234567890ab` esempio. Il preside uscente è il `adminRole` ruolo. La concessione utilizza il vincolo `EncryptionContextSubset` grant per consentire questa autorizzazione solo quando il contesto di crittografia nella `decrypt` richiesta include la `"Department": "IT"` coppia chiave-valore.

```
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, utilizzare il comando. `list-grants`

Per ulteriori informazioni, consulta Grants AWS KMS nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere CreateGrantin AWS CLI Command Reference.

Java

SDKper 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.

```
   /**
    * 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 API i dettagli, vedi CreateGrantin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operationOb = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operationOb)

    val request =
        CreateGrantRequest {
            keyId = keyIdVal
            granteePrincipal = granteePrincipalVal
```

```
            operations = grantOperationList
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.createGrant(request)
        return response.grantId
    }
}
```

- Per API i dettagli, vedi il riferimento CreateGrant AWSSDKa Kotlin API.

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 $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 API i dettagli, vedi CreateGrantin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere CreateGrantPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`CreateKey`** con un AWS SDK o CLI

I seguenti esempi di codice 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 nel seguente esempio di codice:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi CreateKeyin AWS SDK for .NET APIReference.

CLI

AWS CLI

Esempio 1: creare una KMS chiave gestita dal cliente in AWS KMS

L'`create-key`esempio seguente crea una chiave di crittografia KMS simmetrica.

Per creare la KMS chiave di base, una chiave di crittografia simmetrica, non è necessario specificare alcun parametro. I valori predefiniti per tali parametri creano una chiave di crittografia simmetrica.

Poiché questo comando non specifica una politica di chiave, la chiave ottiene la politica di KMS chiave predefinita per le chiavi create a livello di codice. KMS Per visualizzare la politica chiave, utilizzare il `get-key-policy` comando. Per modificare la politica chiave, usa il `put-key-policy` comando.

```
aws kms create-key
```

Il `create-key` comando restituisce i metadati della chiave, incluso l'ID della chiave e ARN della nuova KMS chiave. È possibile utilizzare questi valori per identificare la KMS chiave in altre AWS KMS operazioni. L'output non include i tag. Per visualizzare i tag di una KMS chiave, utilizzare il`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",
        "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 `create-key` comando non consente di specificare un alias. Per creare un alias per la nuova KMS chiave, usa il `create-alias` comando.

Per ulteriori informazioni, vedere Creating keys nella AWS Key Management Service Developer Guide.

Esempio 2: creare una RSA KMS chiave asimmetrica per la crittografia e la decrittografia

L'`create-key`esempio seguente crea una KMS chiave che contiene una coppia di chiavi asimmetrica per RSA la crittografia e la decrittografia.

```
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, vedere Asymmetric keys nella AWS KMS Key Management Service Developer Guide.AWS

Esempio 3: creare una chiave a curva ellittica asimmetrica per la firma e la verifica KMS

Per creare una chiave asimmetrica che contenga una coppia di KMS chiavi asimmetrica ellittica curva (ECC) per la firma e la verifica. Il `--key-usage` parametro è obbligatorio anche se `SIGN_VERIFY` è l'unico valore valido per le chiavi. ECC KMS

```
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 la sezione Chiavi asimmetriche AWS KMS nella AWS Key Management Service Developer Guide.

Esempio 4: Per creare una chiave HMAC KMS

L'`create-key`esempio seguente crea una chiave a 384 bit HMACKMS. Il `GENERATE_VERIFY_MAC` valore del `--key-usage` parametro è obbligatorio anche se è l'unico valore valido per HMAC KMS le chiavi.

```
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 HMACle chiavi AWS KMS nella AWS Key Management Service Developer Guide.

Esempio 4: Per creare una chiave primaria KMS multiregionale

L'`create-key`esempio seguente crea una chiave di crittografia simmetrica primaria multiregione. Poiché i valori predefiniti per tutti i parametri creano una chiave di crittografia simmetrica, per questa chiave è necessario solo il `--multi-region` parametro. KMS 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",
        "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 Asymmetric keys AWS KMS nella AWS Key Management Service Developer Guide.

Esempio 5: creare una KMS chiave per il materiale chiave importato

L'`create-key`esempio seguente crea una KMS chiave senza materiale chiave. Una volta completata l'operazione, è possibile importare il proprio materiale chiave nella KMS chiave. Per creare questa KMS chiave, impostate il `--origin` parametro 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 AWS KMS chiavi nella AWS Key Management Service Developer Guide.

Esempio 6: per creare una KMS chiave in un archivio di HSM chiavi AWS Cloud

L'`create-key`esempio seguente crea una KMS chiave nell'archivio di HSM chiavi AWS Cloud specificato. L'operazione crea la KMS chiave e i relativi metadati AWS KMS e crea il materiale chiave nel HSM cluster AWS Cloud associato all'archivio di 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",
        "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": "AWS_CLOUDHSM"
    }
```

```
}
```

Per ulteriori informazioni, consulta AWS Cloud HSM key stores nella Key Management Service Developer Guide.AWS

Esempio 7: creare una KMS chiave in un archivio di chiavi esterno

L'`create-key`esempio seguente crea una KMS chiave nell'archivio chiavi esterno specificato. I `--xks-key-id` parametri `--custom-key-store-id--origin`, e sono obbligatori in questo comando.

Il `--xks-key-id` parametro specifica l'ID di una chiave di crittografia simmetrica esistente nel gestore di chiavi esterno. Questa chiave funge da materiale chiave esterno per la KMS chiave. Il valore del `--origin` parametro deve essere `EXTERNAL_KEY_STORE` .Il `custom-key-store-id` parametro deve identificare un archivio chiavi esterno collegato al relativo proxy di archiviazione 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",
        "CustomKeyStoreId": "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 la sezione Archivi di chiavi esterni nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere CreateKeyin AWS CLI Command Reference.

Java

SDKper 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.

```
    /**
     * 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 API i dettagli, vedi CreateKeyin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun createKey(keyDesc: String?): String? {
    val request =
        CreateKeyRequest {
            description = keyDesc
            customerMasterKeySpec = CustomerMasterKeySpec.SymmetricDefault
            keyUsage = KeyUsageType.fromValue("ENCRYPT_DECRYPT")
        }

    KmsClient { 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 API i dettagli, vedi il riferimento CreateKey AWSSDKa Kotlin API.

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 $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 API i dettagli, vedi CreateKeyin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere CreateKeyPython (Boto3) Reference.AWS SDK API

Ruby

SDKper Ruby

> **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.

```
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 API i dettagli, vedi CreateKeyin AWS SDK for Ruby APIReference.

Rust

SDKper 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 API i dettagli, CreateKeyconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`Decrypt`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Esempio 1: decrittografare un messaggio crittografato con una KMS chiave simmetrica (Linux e macOS)

Il seguente esempio di `decrypt` comando mostra il metodo consigliato per decrittografare i dati con. AWS CLI Questa versione mostra come decrittografare i dati con una chiave simmetrica. KMS

Fornisci il testo cifrato in un file. Nel valore del `--ciphertext-blob` parametro, usa il `fileb://` prefisso, che indica loro di leggere i dati CLI da un file binario. Se il file non si trova nella directory corrente, digitate il percorso completo del file. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei parametri da un file, consultate Loading AWS CLI parameters from a file < https://docs.aws.amazon.com/cli/ latest/userguide/cli - usage-parameters-file .html> nella AWS Command Line Interface User Guide e 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 KMS chiave per decrittografare il CipherText. Il `--key-id` parametro non è necessario quando si esegue la decrittografia con una chiave simmetrica. KMS AWS KMSpuò ottenere l'ID della KMS chiave utilizzata per crittografare i dati dai metadati nel testo cifrato. Ma è sempre consigliabile specificare la KMS chiave che si sta utilizzando. Questa pratica garantisce l'utilizzo della KMS chiave desiderata e impedisce di decifrare inavvertitamente un testo cifrato utilizzando una KMS chiave non attendibile.Richiedete l'output di testo non crittografato come valore di testo.Il `--query` parametro indica loro di ottenere solo il valore del campo dall'output. CLI `Plaintext` Il `--output` parametro restituisce l'output come text.base64: decodifica il testo semplice e lo salva in un file. L'esempio seguente invia (|) il valore del `Plaintext` parametro 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 `decrypt` comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, vedete Decrypt nel Key Management Service Reference.AWS API

Esempio 2: per decrittografare un messaggio crittografato con una chiave simmetrica KMS (prompt dei comandi di Windows)

L'esempio seguente è lo stesso del precedente, tranne per il fatto che utilizza l'`certutil`utilità per decodificare in Base64 i dati in chiaro. 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, vedete Decrypt nel AWS Key Management Service API Reference.

Esempio 3: decrittografare un messaggio crittografato con una chiave asimmetrica KMS (Linux e macOS)

Il seguente esempio di `decrypt` comando mostra come decrittografare i dati crittografati con una chiave asimmetrica. RSA KMS

Quando si utilizza una KMS chiave asimmetrica, è obbligatorio il `encryption-algorithm` parametro, che specifica l'algoritmo utilizzato per crittografare il testo in chiaro.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID di chiave valido del tuo account. AWS

```
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 `decrypt` comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, consulta la sezione Chiavi asimmetriche nella AWS KMS Key Management Service Developer Guide.AWS

- Per API i dettagli, consulta Decrypt in Command Reference.AWS CLI

Java

SDKper 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 API i dettagli, consulta Decrypt in AWS SDK for Java 2.x API Reference.

Kotlin

SDKper Kotlin

> **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.

```
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 { 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?,
    path: String,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
```

```
        val myVal = decryptResponse.plaintext

        // Write the decrypted data to a file.
        if (myVal != null) {
            File(path).writeBytes(myVal)
        }
    }
}
```

- Per API i dettagli, vedi Decrypt in AWS SDKfor API Kotlin 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 $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 API i dettagli, consulta Decrypt in AWS SDK for PHP API Reference.

Python

SDKper Python (Boto3)

> **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.

```
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: str) -> bytes:
        """
        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"
            ]
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- Per API i dettagli, consulta Decrypt in for AWS SDKPython (Boto3) Reference. API

Ruby

SDKper Ruby

> **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.

```
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 API i dettagli, consulta Decrypt in AWS SDK for Ruby API Reference.

Rust

SDKper 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 API i dettagli, consulta Decrypt in AWS SDKfor Rust API reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori 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 SDK versioni precedenti.

Utilizzare **`DeleteAlias`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAlias`.

CLI

AWS CLI

Per eliminare un alias AWS KMS

L'`delete-alias`esempio seguente elimina l'alias. `alias/example-alias` Il nome dell'alias deve iniziare con alias/.

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

Questo comando non produce alcun output. Per trovare l'alias, usa il comando. `list-aliases`

Per ulteriori informazioni, vedere Eliminazione di un alias nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere DeleteAliasin AWS CLI Command Reference.

Java

SDKper 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.

```
    /**
     * 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 API i dettagli, vedi DeleteAliasin AWS SDK for Java 2.x APIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi DeleteAliasin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere DeleteAliasPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`DescribeKey`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi DescribeKeyin AWS SDK for .NET APIReference.

CLI

AWS CLI

Esempio 1: per trovare informazioni dettagliate su una KMS chiave

L'`describe-key`esempio 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 KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore per il nome di un alias, ma è possibile utilizzare un ID di chiave, una chiaveARN, un nome alias o un alias ARN in questo comando.

```
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",
        "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, vedere Viewing keys nella AWS Key Management Service Developer Guide.

Esempio 2: per ottenere dettagli su una chiave RSA asimmetrica KMS

L'`describe-key`esempio seguente ottiene informazioni dettagliate su una RSA KMS chiave 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: per ottenere dettagli su una chiave di replica multiregionale

L'`describe-key`esempio seguente ottiene i metadati per una chiave di replica multiregionale. Questa chiave multiregionale è una chiave di crittografia simmetrica. L'output di un `describe-key` comando per qualsiasi chiave multiregionale 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",
        "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: per ottenere dettagli su una chiave HMAC KMS

L'`describe-key`esempio seguente ottiene informazioni dettagliate su una HMAC KMS chiave.

```
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 API i dettagli, vedere DescribeKeyin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi DescribeKeyin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun describeSpecifcKey(keyIdVal: String?) {
    val request =
        DescribeKeyRequest {
            keyId = keyIdVal
        }

    KmsClient { 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 API i dettagli, vedi il riferimento DescribeKey AWSSDKa Kotlin API.

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
     * @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 API i dettagli, vedi DescribeKeyin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere DescribeKeyPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`DisableKey`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi DisableKeyin AWS SDK for .NET APIReference.

CLI

AWS CLI

Per disattivare temporaneamente una KMS chiave

L'esempio seguente utilizza il `disable-key` comando per disabilitare una KMS chiave gestita dal cliente. Per riattivare la KMS chiave, utilizzare il `enable-key` comando.

```
aws kms disable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere Enabling and Disabling Keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere DisableKeyin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi DisableKeyin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun disableKey(keyIdVal: String?) {
    val request =
        DisableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        kmsClient.disableKey(request)
        println("$keyIdVal was successfully disabled")
    }
}
```

- Per API i dettagli, vedi il riferimento DisableKey AWSSDKa Kotlin API.

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
     * @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 API i dettagli, vedi DisableKeyin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere DisableKeyPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`EnableKey`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi EnableKeyin AWS SDK for .NET APIReference.

CLI

AWS CLI

Per abilitare una KMS chiave

L'`enable-key`esempio seguente abilita una chiave gestita dal cliente. È possibile utilizzare un comando come questo per abilitare una KMS chiave che è stata temporaneamente disabilitata utilizzando il `disable-key` comando. Puoi anche usarlo per abilitare una KMS chiave che è disabilitata perché era stata pianificata per l'eliminazione e l'eliminazione è stata annullata.

Per specificare la KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o un ARN valore chiave in questo comando.

Prima di eseguire questo comando, sostituite l'ID della chiave di esempio con uno valido.

```
aws kms enable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per verificare che la KMS chiave sia abilitata, utilizzate il `describe-key` comando. Visualizza i valori dei `Enabled` campi `KeyState` e nell'`describe-key`output.

Per ulteriori informazioni, vedere Enabling and Disabling Keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere EnableKeyin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi EnableKeyin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun enableKey(keyIdVal: String?) {
    val request =
        EnableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        kmsClient.enableKey(request)
        println("$keyIdVal was successfully enabled.")
    }
}
```

- Per API i dettagli, vedi il riferimento EnableKey AWSSDKa Kotlin API.

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
     * @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 API i dettagli, vedi EnableKeyin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere EnableKeyPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`EnableKeyRotation`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`EnableKeyRotation`.

CLI

AWS CLI

Per abilitare la rotazione automatica di una chiave KMS

L'`enable-key-rotation`esempio seguente abilita la rotazione automatica di una KMS chiave gestita dal cliente con un periodo di rotazione di 180 giorni. La KMS chiave verrà

ruotata di un anno (circa 365 giorni) dalla data di completamento di questo comando e successivamente ogni anno.

Il `--key-id` parametro identifica la chiave. KMS Questo esempio utilizza un ARN valore chiave, ma è possibile utilizzare l'ID della chiave o quello ARN della KMS chiave. Il `--rotation-period-in-days` parametro specifica il numero di giorni tra ogni data di rotazione. Specificate un valore compreso tra 90 e 2560 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 che la KMS chiave sia abilitata, usa il `get-key-rotation-status` comando.

Per ulteriori informazioni, consulta Rotating keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere EnableKeyRotationin AWS CLI Command Reference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere EnableKeyRotationPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`Encrypt`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Esempio 1: cifrare il contenuto di un file su Linux o macOS

Il `encrypt` comando seguente illustra il metodo consigliato per crittografare i dati con. 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 `--plaintext` parametro per indicare i dati da crittografare. Il valore del parametro deve essere codificato in base 64. Il valore del `plaintext` parametro deve essere codificato in base 64 oppure è necessario utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal file.Se il AWS CLI file non si trova nella directory corrente, digitate il percorso completo del file. Ad esempio: `fileb:///var/tmp/ExamplePlaintextFile` o `fileb://C:\Temp\ExamplePlaintextFile`. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei parametri da un file, vedere Loading Parameters from a File nella AWS Command Line Interface User Guide e Best Practices for Local File `--query` Parameters nel blog AWS `--output` Command Line Tool. Utilizza i parametri 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 dell'output Output dei comandi 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 base64. È necessario decodificare questo testo prima di poterlo utilizzare AWS CLI per decrittografarlo. Salva il testo cifrato binario in un file. La parte finale del comando () salva il testo cifrato binario in un file per facilitarne la decrittografia. `> ExampleEncryptedFile` Per un comando di esempio che utilizza il per decrittografare i dati, consultate gli esempi di decrittografia. AWS CLI

Esempio 2: utilizzo di per AWS CLI crittografare i dati in Windows

Questo esempio è lo stesso del precedente, tranne per il fatto che utilizza lo `certutil` strumento anzichébase64. 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 asimmetrica KMS

Il `encrypt` comando seguente mostra come crittografare il testo in chiaro con una chiave asimmetrica. KMS Il parametro `--encryption-algorithm` è obbligatorio. Come in tutti `encrypt` CLI i comandi, il `plaintext` parametro deve essere codificato in base64 oppure è necessario utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal file. AWS CLI

```
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 i API dettagli, consulta Encrypt in Command Reference.AWS CLI

Java

SDKper 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.

```
   /**
    * 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 API i dettagli, consulta Encrypt in AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
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 { 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?,
    path: String,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Write the decrypted data to a file.
        if (myVal != null) {
            File(path).writeBytes(myVal)
        }
    }
}
```

- Per API i dettagli, vedi Encrypt in AWS SDKfor Kotlin reference API.

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 API i dettagli, consulta Encrypt in AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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) -> str:
        """
        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 API i dettagli, consulta Encrypt in AWS SDKfor Python (APIBoto3) Reference.

Ruby

SDKper Ruby

> **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.

```
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 API i dettagli, consulta Encrypt in AWS SDK for Ruby APIReference.

Rust

SDKper 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 API i dettagli, consulta Encrypt in AWS SDKfor Rust API reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`GenerateDataKey`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateDataKey`.

CLI

AWS CLI

Esempio 1: generare una chiave dati simmetrica a 256 bit

L'`generate-data-key`esempio seguente richiede una chiave dati simmetrica a 256 bit da utilizzare all'esterno di. AWS Il comando restituisce una chiave di dati in testo semplice per l'uso e l'eliminazione immediati e una copia di tale chiave dati crittografata con la chiave specificata. KMS Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

Per richiedere una chiave dati a 256 bit, utilizzate il `key-spec` parametro con un valore di. `AES_256` Per richiedere una chiave dati a 128 bit, utilizzate il `key-spec` parametro con un valore di. `AES_128` Per tutte le altre lunghezze delle chiavi dati, utilizzate il `number-of-bytes` parametro.

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore specifico della KMS chiave 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",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

La `Plaintext` (chiave dati in chiaro) e la `CiphertextBlob` (chiave dati crittografata) vengono restituite in formato con codifica base64.

Per ulteriori informazioni, consulta Data keys < https://docs.aws.amazon.com/kms/ latest/ developerguide/concepts .html #data -keys nella Key Management Service Developer Guide.AWS

Esempio 2: generare una chiave dati simmetrica a 512 bit

L'`generate-data-key`esempio seguente richiede una chiave dati simmetrica a 512 bit per la crittografia e la decrittografia. Il comando restituisce una chiave di dati in testo semplice per l'uso e l'eliminazione immediati e una copia di tale chiave dati crittografata con la chiave specificata. KMS 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, utilizzate il parametro. `number-of-bytes` Per richiedere una chiave dati a 512 bit, l'esempio seguente utilizza il `number-of-bytes` parametro con un valore di 64 (byte).

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore specifico della KMS chiave pari a _. SYMMETRIC DEFAULT

NOTE: I valori nell'output di questo esempio vengono troncati per essere visualizzati.

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

Output:

```
{
```

```
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

`Plaintext`(chiave dati in chiaro) e `CiphertextBlob` (chiave dati crittografata) vengono restituite in formato con codifica base64.

Per ulteriori informazioni, consulta Data keys < https://docs.aws.amazon.com/kms/ latest/ developerguide/concepts .html #data -keys nella Key Management Service Developer Guide.AWS

- Per API i dettagli, vedere GenerateDataKeyin AWS CLI Command Reference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere GenerateDataKeyPython (Boto3) Reference.AWS SDK API

Rust

SDKper 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 API i dettagli, GenerateDataKeyconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`GenerateDataKeyWithoutPlaintext`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateDataKeyWithoutPlaintext`.

CLI

AWS CLI

Per generare una chiave dati simmetrica a 256 bit senza una chiave di testo semplice

L'`generate-data-key-without-plaintext`esempio 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, usa il `key-spec` parametro con un valore di. `AES_256` Per richiedere una chiave dati a 128 bit, utilizzate il `key-spec` parametro con un

valore di. AES_128 Per tutte le altre lunghezze delle chiavi dati, utilizzate il `number-of-bytes` parametro.

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore specifico della KMS chiave 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"
}
```

La `CiphertextBlob` (chiave dati crittografata) viene restituita in formato con codifica base64.

Per ulteriori informazioni, consulta Data keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere GenerateDataKeyWithoutPlaintextin AWS CLI Command Reference.

Rust

SDKper 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 API i dettagli, GenerateDataKeyWithoutPlaintextconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`GenerateRandom`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GenerateRandom`.

CLI

AWS CLI

Esempio 1: generare una stringa di byte casuali a 256 bit (Linux o) macOs

L'`generate-random`esempio seguente genera una stringa di byte casuali a 256 bit (32 byte) con codifica in base64. L'esempio decodifica la stringa di byte e la salva in un file casuale.

Quando si esegue questo comando, è necessario utilizzare il `number-of-bytes` parametro per specificare la lunghezza del valore casuale in byte.

Non si specifica una KMS chiave quando si esegue questo comando. La stringa di byte casuale non è correlata a nessuna KMS chiave.

Per impostazione predefinita, AWS KMS genera il numero casuale. Tuttavia, se si specifica un archivio di chiavi personalizzato< https://docs.aws.amazon.com/kms/ latest/developerguide/ custom - key-store-overview .html>, la stringa di byte casuale viene generata nel HSM cluster AWS Cloud associato all'archivio di chiavi personalizzato.

Questo esempio utilizza i seguenti parametri e valori:

Utilizza il `--number-of-bytes` parametro richiesto con un valore pari `32` a per richiedere una stringa da 32 byte (256 bit). Utilizza il `--output` parametro con valore `text` per indirizzare AWS CLI a restituire l'output come testo, anziché. Utilizza il `--query parameter` per estrarre il valore della `Plaintext` proprietà dalla response.IT invia (|) l'output del comando all'`base64`utilità, che decodifica l'output estratto.Utilizza l'operatore di reindirizzamento (>) per salvare la stringa di byte decodificata in File.it utilizza il JSON `ExampleRandom` operatore di reindirizzamento (>) 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 GenerateRandomnel AWS Key Management Service Reference. API

Esempio 2: generare un numero casuale a 256 bit (prompt dei comandi di Windows)

L'esempio seguente utilizza il `generate-random` comando per generare una stringa di byte casuali a 256 bit (32 byte) con codifica 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'`certutil`utilità di Windows per decodificare in base64 la stringa di byte casuale prima di salvarla in un file.

Innanzitutto, genera una stringa di byte casuali codificata in base64 e la salva in un file temporaneo,. `ExampleRandom.base64`

```
aws kms generate-random \
    --number-of-bytes 32 \
```

```
    --output text \
    --query Plaintext > ExampleRandom.base64
```

Poiché l'output del `generate-random` comando viene salvato in un file, questo esempio non produce alcun output.

Ora utilizzate il `certutil -decode` comando per decodificare la stringa di byte codificata in 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 GenerateRandomnel AWS Key Management Service API Reference.

- Per API i dettagli, vedere GenerateRandomin AWS CLI Command Reference.

Rust

SDKper 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 API i dettagli, GenerateRandomconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`GetKeyPolicy`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Per copiare una politica chiave da una KMS chiave a un'altra KMS chiave

L'`get-key-policy`esempio seguente ottiene la politica chiave da una KMS chiave e la salva in un file di testo. Quindi, sostituisce la policy di una KMS chiave diversa utilizzando il file di testo come input della policy.

Poiché il `--policy` parametro di `put-key-policy` richiede una stringa, è necessario utilizzare l'`--output text`opzione per restituire l'output come stringa di JSON testo anziché.

```
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, vedere PutKeyPolicynel AWS KMSAPIReference.

- Per API i dettagli, vedere GetKeyPolicyin AWS CLI Command Reference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere GetKeyPolicyPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ListAliases`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi ListAliasesin AWS SDK for .NET APIReference.

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 gestite e alle KMS chiavi AWS gestite KMS 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": "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: per elencare tutti gli alias per una chiave particolare KMS

L'esempio seguente utilizza il `list-aliases` comando e il relativo `key-id` parametro per elencare tutti gli alias associati a una chiave particolareKMS.

Ogni alias è associato a una sola KMS chiave, ma una KMS chiave può avere più alias. Questo comando è molto utile perché la AWS KMS console elenca solo un alias per ogni chiave. KMS Per trovare tutti gli alias di una KMS chiave, è necessario utilizzare il `list-aliases` comando.

Questo esempio utilizza l'ID della KMS chiave per il `--key-id` parametro, ma in questo comando è possibile utilizzare un ID chiave, una chiaveARN, un nome alias o un aliasARN.

```
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, vedere Working with Aliases nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere ListAliasesin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi ListAliasesin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun listAllAliases() {
    val request =
        ListAliasesRequest {
            limit = 15
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listAliases(request)
        response.aliases?.forEach { alias ->
            println("The alias name is ${alias.aliasName}")
        }
    }
}
```

- Per API i dettagli, vedi il riferimento ListAliases AWSSDKa Kotlin API.

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 $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 API i dettagli, vedi ListAliasesin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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)?
"
                )
                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
```

- Per API i dettagli, vedere ListAliasesPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ListGrants`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi ListGrantsin AWS SDK for .NET APIReference.

CLI

AWS CLI

Per visualizzare le sovvenzioni su una chiave AWS KMS

L'`list-grants`esempio seguente mostra tutte le concessioni sulla KMS chiave AWS gestita specificata per Amazon DynamoDB nel tuo account. Questa concessione consente a DynamoDB di utilizzare KMS la chiave per conto dell'utente per crittografare una tabella DynamoDB prima di scriverla su disco. È possibile utilizzare un comando come questo per visualizzare le concessioni relative alle chiavi gestite e alle KMS chiavi AWS gestite dal cliente nell'account e nella regioneKMS. AWS

Questo comando utilizza il `key-id` parametro con un ID chiave per identificare la KMS chiave. È possibile utilizzare un ID o una chiave ARN per identificare la KMS chiave. Per ottenere l'ID della chiave o la chiave ARN di una KMS chiave AWS gestita, usa il `list-aliases` comando `list-keys` o.

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

L'output mostra che la concessione autorizza Amazon DynamoDB a utilizzare KMS la chiave per operazioni crittografiche e consente di visualizzare i dettagli sulla chiave `DescribeKey` () e di ritirare KMS le sovvenzioni (). `RetireGrant` Il `EncryptionContextSubset` vincolo limita queste autorizzazioni alle richieste che includono le coppie di contesto 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 AWS KMS nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere ListGrantsin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi ListGrantsin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun displayGrantIds(keyIdVal: String?) {
    val request =
```

```
        ListGrantsRequest {
            keyId = keyIdVal
            limit = 15
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listGrants(request)
        response.grants?.forEach { grant ->
            println("The grant Id is ${grant.grantId}")
        }
    }
}
```

- Per API i dettagli, vedi il riferimento ListGrants AWSSDKa Kotlin API.

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
     * @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 API i dettagli, vedi ListGrantsin AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere ListGrantsPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ListKeyPolicies`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ListKeyPolicies`.

CLI

AWS CLI

Per ottenere i nomi delle politiche chiave per una KMS chiave

L'`list-key-policies`esempio seguente ottiene i nomi delle politiche 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 dal cliente.

Poiché l'unico nome di policy chiave valido è`default`, questo comando non è utile.

Per specificare la KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando.

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
    "PolicyNames": [
    "default"
    ]
}
```

Per ulteriori informazioni sulle politiche AWS KMS chiave, vedere Using Key Policies AWS KMS nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere ListKeyPoliciesin AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi ListKeyPoliciesin AWS SDK for Java 2.x APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere ListKeyPoliciesPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ListKeys`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

.NET

AWS SDK for .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 API i dettagli, vedi ListKeys AWS SDK for .NETAPIReference.

CLI

AWS CLI

Per ottenere le KMS chiavi in un account e in una regione

L'`list-keys`esempio seguente ottiene le KMS chiavi 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, vedere Viewing Keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere ListKeysin AWS CLI Command Reference.

Java

SDKper 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.

```
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 API i dettagli, vedi ListKeys AWS SDK for Java 2.xAPIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun listAllKeys() {
    val request =
        ListKeysRequest {
            limit = 15
        }

    KmsClient { 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 API i dettagli, vedi il riferimento ListKeys AWSSDKa Kotlin API.

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.

```
    /***
```

```
     * @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 API i dettagli, vedi ListKeys AWS SDK for PHPAPIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere ListKeysPython (Boto3) Reference.AWS SDK API

Rust

SDKper 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 API i dettagli, ListKeysconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`PutKeyPolicy`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`PutKeyPolicy`.

CLI

AWS CLI

Modificare la politica chiave per una KMS chiave

L'`put-key-policy`esempio seguente modifica la politica chiave per una chiave gestita dal cliente.

Per iniziare, create una policy chiave e salvatela in un JSON file locale. In questo esempio, il file è`key_policy.json`. È inoltre possibile specificare la politica chiave come valore di stringa del `policy` parametro.

La prima dichiarazione di questa politica chiave autorizza l' AWS account a utilizzare IAM le politiche per controllare l'accesso alla KMS chiave. La seconda istruzione fornisce all'`test-user`utente il permesso di eseguire i `list-keys` comandi `describe-key` and sulla KMS chiave.

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 KMS chiave, in questo esempio viene utilizzato l'ID della chiave, ma è possibile utilizzare anche una chiaveARN. Per specificare la politica della chiave, il comando utilizza il `policy` parametro. Per indicare che la politica è contenuta in un file, utilizza il `file://` prefisso richiesto. Questo prefisso è necessario per identificare i file su tutti i sistemi operativi supportati. Infine, il comando utilizza il `policy-name` parametro con un valore di`default`. Se non viene specificato alcun nome di policy, 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 efficace, utilizzare il `get-key-policy` comando. Il comando di esempio seguente ottiene la politica della chiave per la stessa KMS chiave. Il `output` parametro con un valore di `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, vedere Modifica di una politica AWS chiave nella Key Management Service Developer Guide.

- Per API i dettagli, vedere PutKeyPolicyin 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 API i dettagli, vedi PutKeyPolicy AWS SDK for PHPAPIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere PutKeyPolicyPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ReEncrypt`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`ReEncrypt`.

CLI

AWS CLI

Esempio 1: ricrittografare un messaggio crittografato con una KMS chiave simmetrica diversa (Linux e macOS).

Il seguente esempio di `re-encrypt` comando mostra il metodo consigliato per ricrittografare i dati con. AWS CLI

Fornite il testo cifrato in un file. Nel valore del `--ciphertext-blob` parametro, utilizzate il `fileb://` prefisso, che indica loro di leggere i dati CLI da un file binario. Se il file non si trova nella directory corrente, digitate il percorso completo del file. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei parametri da un file, consultate Loading AWS CLI parameters from a file < https://docs.aws.amazon.com/cli/ latest/userguide/cli - usage-parameters-file .html> nella AWS Command Line Interface User Guide e 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 KMS chiave di origine, che decrittografa il CipherText.Il `--source-key-id` parametro non è richiesto quando si esegue la decrittografia con crittografia simmetrica KMSchiavi. AWS KMSpuò ottenere la KMS chiave utilizzata per crittografare i dati dai metadati nel blob di testo cifrato. Ma è sempre consigliabile specificare la chiave che si sta utilizzando. KMS Questa pratica garantisce l'utilizzo della KMS chiave desiderata e impedisce di decrittografare inavvertitamente un testo cifrato utilizzando una KMS chiave non attendibile.Specificate la chiave di destinazione, che cripta nuovamente i dati.Il parametro KMS è sempre obbligatorio. `--destination-key-id` Questo esempio

utilizza una chiaveARN, ma è possibile utilizzare qualsiasi identificatore di chiave valido. Richiedere l'output in chiaro come valore di testo. Il `--query` parametro indica loro di CLI ottenere solo il valore del `Plaintext` campo dall'output. Il `--output` parametro restituisce l'output come text.BASE64-Decode il testo non crittografato e lo salva in un file.L'esempio seguente trasferisce (|) il valore del parametro all'utilità Base64, che lo decodifica. `Plaintext` 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 `re-encrypt` comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, vedere ReEncrypt < https://docs.aws.amazon.com/kms/ latest/ APIReference/API _ ReEncrypt .html nel Key Management Service Reference.AWS API

Esempio 2: Per crittografare nuovamente un messaggio crittografato con una KMS chiave simmetrica diversa (prompt dei comandi di Windows).

Il seguente esempio di `re-encrypt` comando è lo stesso del precedente, tranne per il fatto che utilizza l'`certutil`utilità per decodificare in Base64 i dati di testo in chiaro. 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
```

Quindi usa l'`certutil`utilità

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, vedere ReEncrypt < https://docs.aws.amazon.com/kms/ latest/ APIReference/API _ ReEncrypt .html nel AWS Key Management Service API Reference.

- Per API i dettagli, vedere ReEncryptin AWS CLI Command Reference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere ReEncryptPython (Boto3) Reference.AWS SDK API

Ruby

SDKper Ruby

> **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.

```
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 API i dettagli, vedi ReEncrypt AWS SDK for RubyAPIReference.

Rust

SDKper 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 API i dettagli, ReEncryptconsulta AWS SDKRust API Reference.

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`RetireGrant`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`RetireGrant`.

CLI

AWS CLI

Ritirare una sovvenzione relativa a una chiave master del cliente

L'`retire-grant`esempio seguente elimina una concessione da una KMS chiave.

Il comando di esempio seguente specifica i parametri `grant-id` e. `key-id` Il valore del `key-id` parametro deve essere la chiave ARN della KMS chiave.

```
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 che la concessione è stata ritirata, usa il `list-grants` comando.

Per ulteriori informazioni, consulta Ritiro e revoca delle sovvenzioni nella AWS Key Management Service Developer Guide.

- Per i API dettagli, vedere RetireGrantin Command Reference.AWS CLI

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere RetireGrantPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`RevokeGrant`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Per revocare una concessione su una chiave master del cliente

L'`revoke-grant`esempio seguente elimina una concessione da una chiave. KMS Il comando di esempio seguente specifica i parametri `grant-id` e. `key-id` Il valore del `key-id` parametro può essere l'ID o la chiave ARN della KMS chiave.

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per confermare che la concessione è stata revocata, usa il `list-grants` comando.

Per ulteriori informazioni, consulta Ritiro e revoca delle sovvenzioni nella AWS Key Management Service Developer Guide.

- Per i API dettagli, vedere RevokeGrantin Command Reference.AWS CLI

Java

SDKper 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.

```
   /**
    * 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 API i dettagli, vedi RevokeGrant AWS SDK for Java 2.xAPIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi RevokeGrant AWS SDK for PHPAPIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere RevokeGrantPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`ScheduleKeyDeletion`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Per pianificare l'eliminazione di una KMS chiave gestita dal cliente.

L'`schedule-key-deletion`esempio seguente pianifica l'eliminazione della KMS chiave gestita dal cliente specificata in 15 giorni.

Il `--key-id` parametro identifica la KMS chiave. Questo esempio utilizza un ARN valore chiave, ma è possibile utilizzare l'ID della chiave o la ARN KMS chiave. Il `--pending-`

`window-in-days` parametro specifica la durata del periodo di attesa di 7-30 giorni. Per impostazione predefinita, il periodo di attesa è di 30 giorni. Questo esempio specifica il valore 15, che indica di AWS eliminare definitivamente la KMS chiave 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 la chiaveARN, lo stato della chiave, il periodo di attesa (`PendingWindowInDays`) e la data di eliminazione in formato Unix. Per visualizzare la data di cancellazione nell'ora locale, usa la AWS KMS console. KMSle chiavi nello stato della `PendingDeletion` chiave non possono essere utilizzate nelle operazioni crittografiche.

```
{
    "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, vedere Eliminazione delle chiavi nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere ScheduleKeyDeletionin AWS CLI Command Reference.

Java

SDKper 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.

```
    /**
     * 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 API i dettagli, vedi ScheduleKeyDeletion AWS SDK for Java 2.xAPIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi ScheduleKeyDeletion AWS SDK for PHPAPIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere ScheduleKeyDeletionPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`Sign`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Esempio 1: generare una firma digitale per un messaggio

L'`sign`esempio seguente genera una firma crittografica per un breve messaggio. L'output del comando include un `Signature` campo codificato in base 64 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 tua KMS chiave, usa il comando. `describe-key`

Nella AWS CLI versione 2.0, il valore del `message` parametro deve essere codificato in Base64. In alternativa, è possibile salvare il messaggio in un file e utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal AWS CLI file.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account. L'ID della chiave deve rappresentare una KMS chiave asimmetrica con un utilizzo chiave di _. 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+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Per ulteriori informazioni sull'utilizzo delle KMS chiavi asimmetriche in AWS KMS, consulta la sezione Chiavi asimmetriche nella Key Management Service Developer Guide. AWS KMS AWS

Esempio 2: per salvare una firma digitale in un file (Linux e) macOs

L'`sign`esempio seguente genera una firma crittografica per un breve messaggio memorizzato in un file locale. Il comando ottiene anche la `Signature` proprietà dalla risposta, Base64-la decodifica e la salva nel file. ExampleSignature È possibile utilizzare il file della firma in un `verify` comando che verifica la firma.

Il `sign` comando richiede un messaggio con codifica Base64 e un algoritmo di firma supportato dalla chiave asimmetrica. KMS Per ottenere gli algoritmi di firma supportati dalla chiave, usa il comando. KMS `describe-key`

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account. L'ID della chiave deve rappresentare una KMS chiave asimmetrica con un utilizzo chiave di _. 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 `Signature` proprietà dell'output e la salva in un file.

Per ulteriori informazioni sull'utilizzo delle chiavi asimmetriche in AWS KMS, vedete Asymmetric KMS keys AWS KMS nella Key Management Service Developer Guide.AWS

- Per i API dettagli, consulta Sign in Command Reference.AWS CLI

Java

SDKper 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 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 API i dettagli, consulta Sign in AWS SDK for Java 2.x APIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, consulta Sign in AWS SDK for PHP APIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, consulta Sign in AWS SDKfor Python (Boto3) Reference. API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`TagResource`** con un AWS SDK o CLI

I seguenti esempi di codice 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:

- Impara le nozioni di base

CLI

AWS CLI

Per aggiungere un tag a una KMS chiave

L'`tag-resource`esempio seguente aggiunge `"Purpose":"Test"` e `"Dept":"IT"` contrassegna una KMS chiave gestita dal cliente. È possibile utilizzare tag come questi per etichettare KMS le chiavi e creare categorie di KMS chiavi per le autorizzazioni e il controllo.

Per specificare la KMS chiave, utilizzate il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando.

```
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 un AWS KMS KMS tasto, usa il `list-resource-tags` comando.

Per ulteriori informazioni sull'utilizzo dei tag in AWS KMS, consulta Tagging keys nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere TagResourcein AWS CLI Command Reference.

Java

SDKper 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 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 API i dettagli, vedi TagResource AWS SDK for Java 2.xAPIReference.

PHP

SDK per PHP

> **Note**
>
> C'è altro da sapere 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 API i dettagli, vedi TagResource AWS SDK for PHPAPIReference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere TagResourcePython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`UpdateAlias`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateAlias`.

CLI

AWS CLI

Per associare un alias a una chiave diversa KMS

L'`update-alias`esempio seguente associa l'alias a una chiave `alias/test-key` diversa. KMS

Il `--alias-name` parametro specifica l'alias. Il valore del nome alias deve iniziare con `alias/` .Il `--target-key-id` parametro specifica la KMS chiave da associare all'alias. Non è necessario specificare la KMS chiave 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 trovare l'alias, usa il `list-aliases` comando.

Per ulteriori informazioni, vedere Aggiornamento degli alias nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere UpdateAliasin AWS CLI Command Reference.

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, vedere UpdateAliasPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Utilizzare **`Verify`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`Verify`.

CLI

AWS CLI

Per verificare una firma digitale

L'`verify`esempio seguente verifica una firma crittografica per un breve messaggio con codifica Base64. L'ID della chiave, il messaggio, il tipo di messaggio e l'algoritmo di firma devono essere gli stessi utilizzati per firmare il messaggio. La firma specificata non può essere codificata in base 64. Per informazioni sulla decodifica della firma restituita dal comando, consultate gli esempi di `sign` comandi. `sign`

L'output del comando include un `SignatureValid` campo booleano che indica che la firma è stata verificata. Se la convalida della firma fallisce, anche il `verify` comando fallisce.

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 asimmetriche in AWS KMS, vedere Using asimmetric KMS keys nella Key Management Service Developer Guide.AWS

- Per i API dettagli, consulta Verify in Command Reference.AWS CLI

Python

SDKper Python (Boto3)

> **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.

```
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 API i dettagli, consulta Verify in AWS SDKfor Python (Boto3) Reference. API

Per un elenco completo delle guide per gli AWS SDK sviluppatori 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 SDK versioni precedenti.

Attestazione crittografica per AWS Nitro Enclaves

AWS KMS supporta l'attestazione crittografica per Nitro Enclaves.AWS Le applicazioni che supportano AWS Nitro Enclaves eseguono le seguenti operazioni AWS KMS crittografiche con un documento di attestazione firmato per l'enclave. Questi AWS KMS APIs verificano che il documento di attestazione provenga da un'enclave di Nitro. 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.

- Decrypt
- DeriveSharedSecret
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

La tabella seguente mostra in che modo la risposta alle richieste di enclave Nitro differisce dalla risposta standard per ciascuna operazione. API

AWS KMS operazione	Risposta standard	Risposta per AWS Nitro Enclaves
`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 AWS Nitro Enclaves
	(Restituisce anche una copia della chiave dati crittografata da una KMS chiave)	(Restituisce anche una copia della chiave dati crittografata da una KMS chiave)
`GenerateDataKeyPair`	Restituisce una copia in testo normale della chiave privata (Restituisce anche la chiave pubblica e una copia della chiave privata crittografata da una KMS chiave)	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 KMS chiave)
`GenerateRandom`	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 di enclave con una AWS KMS chiave basata sul contenuto del documento di attestazione. Puoi anche monitorare le richieste AWS KMS per la tua enclave Nitro nei tuoi registri. AWS CloudTrail

Ulteriori informazioni

- Attestazione crittografica
- AWS KMS chiavi di condizione per AWS Nitro Enclaves
- Come richiedere un'enclave AWS KMS APIs Nitro
- Richieste di monitoraggio per enclavi Nitro

Come richiedere un'enclave AWS KMS APIs Nitro

AWS KMS APIsPer richiedere un'enclave Nitro, usa il `Recipient` parametro nella richiesta per fornire il documento di attestazione firmato per l'enclave e l'algoritmo di crittografia da utilizzare con la chiave pubblica dell'enclave. 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'enclave Nitro. AWS AWS KMS si affida alla firma digitale del documento di attestazione dell'enclave per dimostrare che la chiave pubblica contenuta nella richiesta proviene da un'enclave valida. Non è possibile fornire il proprio certificato per firmare digitalmente il documento di attestazione.

Per specificare il `Recipient` parametro, usa Nitro Enclaves o uno qualsiasi.AWS SDK AWS SDK Le AWS Nitro EnclavesSDK, supportate solo all'interno di un'enclave Nitro, aggiungono automaticamente il parametro e i relativi valori a ogni richiesta. `Recipient` AWS KMS Per effettuare richieste per le enclavi Nitro in AWS SDKs, devi specificare il parametro e i relativi valori. `Recipient` Il supporto per l'attestazione crittografica dell'enclave Nitro in AWS SDKs è stato introdotto a marzo 2023.

AWS KMS supporta le chiavi relative alle condizioni delle policy che è possibile utilizzare per consentire o negare operazioni di enclave con una AWS KMS chiave basata sul contenuto del documento di attestazione. Puoi anche monitorare le richieste AWS KMS per la tua enclave Nitro nei tuoi registri. AWS CloudTrail

Per informazioni dettagliate sul `Recipient` parametro e sul campo di AWS `CiphertextForRecipient` risposta, consulta Decrypt,,,, e GenerateRandomgli argomenti in AWS Key Management Service APIReference DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair, Nitro Enclaves o altri.AWS SDK AWS SDK Per informazioni sulla configurazione dei dati e delle chiavi dati per la crittografia, consulta Utilizzo dell'attestazione crittografica con. AWS KMS

Richieste di monitoraggio per enclavi Nitro

Puoi usare i tuoi AWS CloudTrail log per monitorare Decrypt,, DeriveSharedSecretGenerateDataKey, GenerateDataKeyPaire le GenerateRandomoperazioni per un'enclave Nitro. AWS In queste voci di registro, il `additionalEventData` campo contiene un `recipient` campo con l'ID del modulo (`attestationDocumentModuleId`), l'image digest () e i registri di configurazione della piattaforma

(`attestationDocumentEnclaveImageDigest`) contenuti nel documento di attestazione contenuto nella richiesta. PCRs Questi campi sono inclusi solo quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato da un'enclave Nitro. AWS

L'ID del modulo è l'ID enclave dell'enclave Nitro. L'image digest è l'hash dell'immagine dell'SHA384enclave. È possibile utilizzare l'image digest e PCR i valori in base alle politiche e alle politiche chiave. IAM Per informazioni suPCRs, consulta Dove reperire le misure di un'enclave nella Guida per l'utente di AWS Nitro Enclaves.

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'GenerateDataKeyoperazione 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": "2014-11-04T00:52:40Z",
    "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'GenerateDataKeyPairoperazione 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'GenerateRandomoperazione 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"
}
```

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 EBS volumi Amazon, 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 AWS KMS, consulta Service Integration.AWS

Negli argomenti seguenti viene illustrato in dettaglio l'utilizzo di determinati servizi AWS KMS, comprese le KMS chiavi che supportano, come gestiscono le chiavi di dati, le autorizzazioni richieste e come tenere traccia dell'utilizzo delle KMS chiavi dell'account da parte di ciascun servizio.

> **Important**
>
> AWS i servizi integrati con AWS KMS utilizzano solo KMS chiavi di crittografia simmetriche per crittografare i dati. Questi servizi non supportano la crittografia con chiavi asimmetriche. KMS Per informazioni su come determinare se una KMS chiave è simmetrica o asimmetrica, vedere. Identifica diversi tipi di chiave

Argomenti

- Come si AWS CloudTrail usa AWS KMS
- Come utilizza Amazon Elastic Block Store (AmazonEBS) AWS KMS
- Come EMR utilizza Amazon AWS KMS
- Come utilizza Amazon Redshift AWS KMS

Come si AWS CloudTrail usa AWS KMS

Puoi utilizzarle per AWS CloudTrail registrare AWS API chiamate e altre attività per conto tuo Account AWS e per salvare le informazioni registrate in file di log in un bucket Amazon Simple

Storage Service (Amazon S3) a tua scelta. Per impostazione predefinita, i file di log CloudTrail inseriti nel bucket S3 sono crittografati utilizzando la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (-S3). SSE Ma puoi scegliere invece di utilizzare la crittografia lato server con una chiave (-). KMS SSE KMS Per informazioni su come crittografare i file di CloudTrail registro con AWS KMS, consulta Encrypting CloudTrail Log Files with AWS KMS keys (SSE-KMS) nella Guida per l'utente.AWS CloudTrail

> **Important**
>
> AWS CloudTrail e Amazon S3 supporta solo sistemi simmetrici. AWS KMS keys Non è possibile utilizzare una chiave asimmetrica KMS per crittografare i log. CloudTrail Per informazioni su come determinare se una KMS chiave è simmetrica o asimmetrica, consulta. Identifica diversi tipi di chiave

Non si paga un costo per l'utilizzo delle chiavi quando si CloudTrail leggono o scrivono file di registro crittografati con una chiave -. SSE KMS Tuttavia, si paga un costo per l'utilizzo delle chiavi quando si accede ai file di CloudTrail registro crittografati con una KMS chiave SSE -. Per informazioni sui AWS KMS prezzi, consulta la sezione AWS Key Management Service Prezzi. Per informazioni sui CloudTrail prezzi, consulta la sezione AWS CloudTrail Prezzi e Gestione dei costi nella Guida AWS CloudTrail per l'utente.

Argomenti

- Capire quando viene utilizzata la KMS chiave

Capire quando viene utilizzata la KMS chiave

La crittografia dei file di CloudTrail registro AWS KMS si basa sulla funzionalità di Amazon S3 denominata crittografia lato server con un (-). AWS KMS key SSE KMS Per ulteriori informazioni su SSE -KMS, consulta Protezione dei dati utilizzando la crittografia lato server con KMS chiavi (SSE-KMS) nella Guida per l'utente di Amazon Simple Storage Service.

Quando configuri AWS CloudTrail per l'usoSSE, KMS per crittografare i tuoi file di registro CloudTrail e Amazon S3 utilizza AWS KMS keys il tuo quando esegui determinate azioni con tali servizi.
Le sezioni seguenti spiegano quando e come tali servizi possono utilizzare la tua KMS chiave e forniscono informazioni aggiuntive che puoi utilizzare per convalidare questa spiegazione.

Azioni che causano CloudTrail l'utilizzo della chiave da parte di Amazon S3 KMS

- Ti configuri CloudTrail per crittografare i file di registro con AWS KMS key
- CloudTrail inserisce un file di registro nel tuo bucket S3
- Hai a disposizione un file di log crittografato dal tuo bucket S3

Ti configuri CloudTrail per crittografare i file di registro con AWS KMS key

Quando aggiorni la CloudTrail configurazione per utilizzare la tua KMS chiave, CloudTrail invia una `GenerateDataKey`richiesta per AWS KMS verificare che la KMS chiave esista e che CloudTrail sia autorizzato a usarla per la crittografia. CloudTrail non utilizza la chiave dati risultante.

La richiesta `GenerateDataKey` include le seguenti informazioni per il contesto di crittografia:

- Il nome della risorsa Amazon (ARN) del CloudTrail percorso
- Il ARN bucket S3 e il percorso in cui vengono consegnati i file di CloudTrail registro

La `GenerateDataKey` richiesta genera una voce nei CloudTrail log simile all'esempio seguente. Quando vedete una voce di registro come questa, potete determinare che CloudTrail (1) ha chiamato AWS KMS (2) `GenerateDataKey` operation (3) per uno specifico trail (4). AWS KMS ha creato la chiave dati con una KMS chiave specifica (5).

> **Note**
>
> Potrebbe essere necessario scorrere verso destra per visualizzare alcune delle didascalie nella seguente voce di log di esempio.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
```

```
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::086441151436:user/
AWSCloudTrail",
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "AWSCloudTrail",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T21:15:33Z"
    }},
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:15:33Z",
  "eventSource":
 "kms.amazonaws.com",
  "eventName":
 "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAliasForCloudTrailKMS
 key",
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/AWSLogs/111122223333/"
    },
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "581f1f11-88b9-11e5-9c9c-595a1fb59ac0",
  "eventID": "3cdb2457-c035-4890-93b6-181832b9e766",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
```

```
}
```

CloudTrail inserisce un file di registro nel tuo bucket S3

Ogni volta che CloudTrail inserisce un file di registro nel tuo bucket S3, Amazon S3 invia `GenerateDataKey`una richiesta AWS KMS a per conto di. CloudTrail In risposta a questa richiesta, AWS KMS genera una chiave dati univoca e quindi invia ad Amazon S3 due copie della chiave dati, una in testo semplice e una crittografata con la chiave specificata. KMS Amazon S3 utilizza la chiave dati in chiaro per crittografare il file di CloudTrail registro e quindi rimuove la chiave dati in testo semplice dalla memoria il prima possibile dopo l'uso. Amazon S3 archivia la chiave dati crittografata come metadati con il file di registro crittografato CloudTrail .

La richiesta `GenerateDataKey` include le seguenti informazioni per il contesto di crittografia:

- Il nome della risorsa Amazon (ARN) del CloudTrail percorso
- L'ARNoggetto S3 (il file di CloudTrail registro)

Ogni `GenerateDataKey` richiesta genera una voce nei CloudTrail log simile all'esempio seguente. Quando vedete una voce di registro come questa, potete determinare che CloudTrail
(1)
ha chiamato l'`GenerateDataKey`operazione AWS KMS
(2) (3)
per uno specifico trail
(4)
per proteggere uno specifico file di registro
(5).
AWS KMS ha creato la chiave dati con la KMS chiave specificata
(6),
mostrata due volte nella stessa voce di registro.

> **Note**
>
> Potrebbe essere necessario scorrere verso destra per visualizzare alcune delle didascalie nella seguente voce di log di esempio.

```
{
```

```
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:i-34755b85",
    "arn": "arn:aws:sts::086441151436:assumed-role/AWSCloudTrail/
i-34755b85",(1)
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-11-11T20:45:25Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::086441151436:role/AWSCloudTrail",
        "accountId": "086441151436",
        "userName": "AWSCloudTrail"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:15:58Z",
  "eventSource":
 "kms.amazonaws.com",(2)
  "eventName":
 "GenerateDataKey",(3)
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",(4)
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"(5)
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(6)
    "keySpec": "AES_256"
```

```
  },
  "responseElements": null,
  "requestID": "66f3f74a-88b9-11e5-b7fb-63d925c72ffe",
  "eventID": "7738554f-92ab-4e27-83e3-03354b1aa898",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(6)
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

Hai a disposizione un file di log crittografato dal tuo bucket S3

Ogni volta che ricevi un file di CloudTrail registro crittografato dal tuo bucket S3, Amazon S3 invia `Decrypt`una richiesta AWS KMS a tuo nome per decrittografare la chiave dati crittografata del file di registro. In risposta a questa richiesta, AWS KMS utilizza la tua KMS chiave per decrittografare la chiave dati e quindi invia la chiave dati in testo semplice ad Amazon S3. Amazon S3 utilizza la chiave dati in testo semplice per decrittografare il file di CloudTrail registro e quindi rimuove la chiave dati in testo semplice dalla memoria il prima possibile dopo l'uso.

La richiesta `Decrypt` include le seguenti informazioni per il contesto di crittografia:

- Il nome della risorsa Amazon (ARN) del CloudTrail percorso
- L'ARNoggetto S3 (il file di CloudTrail registro)

Ogni `Decrypt` richiesta genera una voce nei CloudTrail log simile all'esempio seguente. Quando vedi una voce di registro come questa, puoi determinare che un utente in Account AWS
(1)
ha chiamato la AWS KMS
(2)
`Decrypt` operation
(3)
per un trail
(4)
specifico e un file di registro specifico

(5).
AWS KMS ha decrittografato la chiave dati con una KMS chiave specifica
() 6 .

> **Note**
>
> Potrebbe essere necessario scorrere verso destra per visualizzare alcune delle didascalie nella seguente voce di log di esempio.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/cloudtrail-
admin",1
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "cloudtrail-admin",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T20:48:04Z"
    }},
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:20:52Z",
  "eventSource":
 "kms.amazonaws.com",2
  "eventName":
 "Decrypt",3
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",4
```

```
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"5
    }
  },
  "responseElements": null,
  "requestID": "16a0590a-88ba-11e5-b406-436f15c3ac01",
  "eventID": "9525bee7-5145-42b0-bed5-ab7196a16daa",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",6
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Come utilizza Amazon Elastic Block Store (AmazonEBS) AWS KMS

Questo argomento illustra in dettaglio come Amazon Elastic Block Store (AmazonEBS) utilizza AWS KMS per crittografare volumi e snapshot. Per istruzioni di base sulla crittografia EBS dei volumi Amazon, consulta Amazon EBS Encryption.

Argomenti

- EBSCrittografia Amazon
- Utilizzo di KMS chiavi e chiavi dati
- Contesto EBS di crittografia Amazon
- Rilevamento dei guasti di Amazon EBS
- Utilizzo AWS CloudFormation per creare EBS volumi Amazon crittografati

EBSCrittografia Amazon

Quando colleghi un EBS volume Amazon crittografato a un tipo di istanza Amazon Elastic Compute Cloud (AmazonEC2) 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 funzionalità è supportata su tutti i tipi di EBS volume Amazon. 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 EBS volume viene determinato al momento della creazione del 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 EBS volumi e le copie di istantanee nella tua Account AWS regione. Questa impostazione di configurazione non influisce sui volumi o sugli snapshot esistenti. Per maggiori dettagli, consulta Amazon EBS encryption nella Amazon EBS User Guide.

Utilizzo di KMS chiavi e chiavi dati

Quando crei un EBS volume Amazon crittografato, specifichi un AWS KMS key. Per impostazione predefinita, Amazon EBS utilizza Chiave gestita da AWSfor 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, devi EBS autorizzare Amazon a utilizzare la KMS chiave per tuo conto. Per un elenco delle autorizzazioni richieste, consulta Autorizzazioni per IAM gli utenti nella Amazon User Guide o nella Amazon EC2 EC2 User Guide.

> **Important**
>
> Amazon EBS supporta solo chiavi simmetriche KMS. Non è possibile utilizzare una KMSchiave asimmetrica per crittografare un volume Amazon. EBS Per informazioni su come determinare se una KMS chiave è simmetrica o asimmetrica, consulta. Identifica diversi tipi di chiave

Per ogni volume, Amazon EBS chiede AWS KMS di generare una chiave dati univoca crittografata con la KMS chiave specificata. Amazon EBS archivia la chiave dati crittografata con il volume.

Quindi, quando colleghi il volume a un'EC2istanza 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 tutti gli I/O del disco sul volume. Per i dettagli, consulta Come funziona la EBS crittografia nella Amazon EC2 User Guide o Amazon EC2 User Guide.

Contesto EBS di crittografia Amazon

Nelle sue richieste GenerateDataKeyWithoutPlaintexte Decrypt a, 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 le istantanee crittografate create con l'EBSCreateSnapshotoperazione Amazon, Amazon EBS utilizza l'ID del volume come valore del contesto di crittografia. Nel `requestParameters` campo di una voce di CloudTrail registro, il contesto di crittografia è simile al seguente:

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Per le istantanee crittografate create con l'EC2CopySnapshotoperazione Amazon, Amazon EBS utilizza l'ID snapshot come valore del contesto di crittografia. Nel `requestParameters` campo di una voce di CloudTrail registro, il contesto di crittografia è simile al seguente:

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Rilevamento dei guasti di Amazon EBS

Per creare un EBS volume crittografato o collegare il volume a un'EC2istanza, Amazon EBS e l'EC2infrastruttura Amazon devono essere in grado di utilizzare la KMS chiave specificata per la crittografia EBS del volume. Quando la KMS chiave non è utilizzabile, ad esempio quando lo stato della chiave non lo è, la creazione `Enabled` del volume o il relativo allegato falliscono.

In questo caso, Amazon EBS invia un evento ad Amazon EventBridge (in precedenza 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 Amazon EC2 User Guide, 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 correggere questi errori, assicurati che la KMS chiave che hai specificato per la crittografia dei EBS volumi sia abilitata. A tale scopo, visualizzate innanzitutto la KMS chiave per determinarne lo stato corrente (la colonna Status in AWS Management Console). Quindi, consulta le informazioni contenute in uno dei seguenti collegamenti:

- Se lo stato della KMS chiave è disabilitato, abilitalo.
- Se lo stato della KMS chiave è in attesa di importazione, importa il materiale chiave.
- Se lo stato della KMS chiave è in attesa di eliminazione, annulla l'eliminazione della chiave.

Utilizzo AWS CloudFormation per creare EBS volumi Amazon crittografati

Puoi utilizzarlo AWS CloudFormationper creare EBS volumi Amazon crittografati. Per ulteriori informazioni, consulta AWS::EC2: :Volume nella Guida per l'AWS CloudFormation utente.

Come EMR utilizza Amazon AWS KMS

Quando utilizzi un EMR cluster Amazon, puoi configurare il cluster per crittografare i dati inattivi prima di salvarli in una posizione di archiviazione persistente. Puoi crittografare i dati inattivi sul 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. I seguenti argomenti spiegano come un EMR cluster Amazon utilizza una KMS chiave per crittografare i dati inattivi.

> **Important**
>
> Amazon EMR supporta solo chiavi simmetriche KMS. Non è possibile utilizzare una KMSchiave asimmetrica per crittografare i dati inattivi in un cluster Amazon. EMR Per informazioni su come determinare se una KMS chiave è simmetrica o asimmetrica, consulta. Identifica diversi tipi di chiave

EMRI cluster Amazon crittografano anche i dati in transito, il che significa che il cluster crittografa i dati prima di inviarli attraverso la rete. Non è possibile utilizzare una KMS chiave per crittografare i dati in transito. Per ulteriori informazioni, consulta la sezione In-Transit Data Encryption nella Amazon EMR Management Guide.

Per ulteriori informazioni su tutte le opzioni di crittografia disponibili in AmazonEMR, consulta Encryption Options nella Amazon EMR Management Guide.

Argomenti

- Crittografia dei dati sul EMR file system () EMRFS
- Crittografia dei dati su volumi di storage di nodi cluster
- Contesto di crittografia

Crittografia dei dati sul EMR file system () EMRFS

EMRI cluster Amazon utilizzano due file system distribuiti:

- Il file system distribuito Hadoop (). HDFS HDFSla crittografia non utilizza una KMS chiave in. AWS KMS
- Il EMR file system (EMRFS). EMRFSè un'implementazione HDFS che consente ai EMR cluster Amazon di archiviare dati in Amazon Simple Storage Service (Amazon S3). EMRFSsupporta quattro opzioni di crittografia, due delle quali utilizzano una KMS chiave in. AWS KMS Per ulteriori informazioni su tutte e quattro le opzioni di EMRFS crittografia, consulta Encryption Options nella Amazon EMR Management Guide.

Le due opzioni di EMRFS crittografia che utilizzano una KMS chiave utilizzano le seguenti funzionalità di crittografia offerte da Amazon S3:

- Protezione dei dati utilizzando la crittografia lato server con AWS Key Management Service (SSE-). KMS Il EMR cluster Amazon invia dati ad Amazon S3. Amazon S3 utilizza una KMS chiave 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 EMRFS con SSE - KMS.
- Protezione dei dati utilizzando la crittografia lato client (-). CSE KMS I dati in Amazon EMR vengono crittografati con e AWS KMS key prima di essere inviati ad Amazon S3 per l'archiviazione. Per ulteriori informazioni su come effettuare tale operazione, consulta EMRFSProcesso CSE per crittografare i dati con - KMS.

Quando configuri un EMR cluster Amazon per crittografare i dati EMRFS con una KMS chiave, scegli la KMS chiave che desideri venga utilizzata da Amazon S3 o dal cluster EMR Amazon. Con SSE - KMS, puoi scegliere la chiave Chiave gestita da AWS per Amazon S3 con l'alias aws/s3 o una chiave simmetrica gestita dal cliente che crei tu. 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 tuo EMR cluster Amazon sia autorizzato a utilizzare la KMS chiave. Per ulteriori informazioni, consulta Using AWS KMS keys for encryption nella Amazon EMR Management Guide.

Sia per la crittografia lato server che lato client, la KMS chiave scelta è la chiave principale in un flusso di lavoro di crittografia a busta. I dati vengono crittografati con una chiave dati univoca che viene crittografata sotto la chiave in. 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 EMRFS con SSE - KMS
- EMRFSProcesso CSE per crittografare i dati con - KMS

Processo di crittografia dei dati EMRFS con SSE - KMS

Quando configuri un EMR cluster Amazon per utilizzare SSE -KMS, il processo di crittografia funziona in questo modo:

1. Il cluster invia i dati ad Amazon S3 per lo storage in un bucket S3.
2. Amazon S3 invia una GenerateDataKeyrichiesta a AWS KMS, specificando l'ID della KMS chiave che hai scelto quando hai configurato il cluster da utilizzare -. 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 non crittografato) e l'altra copia è crittografata sotto 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 KMS chiave 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.

EMRFSProcesso CSE per crittografare i dati con - KMS

Quando configuri un EMR cluster Amazon per utilizzare CSE -KMS, il processo di crittografia funziona in questo modo:

1. Quando è pronto per archiviare i dati in Amazon S3, il cluster invia una GenerateDataKeyrichiesta a AWS KMS, specificando l'ID chiave della KMS chiave che hai scelto quando hai configurato il cluster da utilizzare -. 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 univoca (chiave dati) e quindi invia due copie di questa chiave di dati al cluster. Una copia non è crittografata (testo non crittografato) e l'altra copia è crittografata sotto 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 dei dati crittografati dall'oggetto crittografato e quindi invia la chiave dei dati crittografati 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 KMS chiave 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 EMR cluster Amazon è una raccolta di istanze Amazon Elastic Compute Cloud (AmazonEC2). Ogni istanza nel cluster viene chiamata nodo cluster o nodo. Ogni nodo può avere due tipi di volumi di storage: volumi di instance store e volumi Amazon Elastic Block Store (AmazonEBS). Puoi configurare il cluster per utilizzare Linux Unified Key Setup (LUKS) per crittografare entrambi i tipi di volumi di storage sui nodi (ma non il volume di avvio di ciascun nodo). Questa si chiama la crittografia dei dati su disco locale.

Quando abiliti la crittografia del disco locale per un cluster, puoi scegliere di crittografare la LUKS chiave con una KMS chiave in ingresso. AWS KMS Devi scegliere una chiave gestita dal cliente che hai creato; non è possibile utilizzare una Chiave gestita da AWS. Se scegli una chiave gestita dal cliente, devi assicurarti che il tuo EMR cluster Amazon sia autorizzato a utilizzare la KMS chiave. Per ulteriori informazioni, consulta Using AWS KMS keys for encryption nella Amazon EMR Management Guide.

Quando abiliti la crittografia del disco locale utilizzando una KMS chiave, il processo di crittografia funziona in questo modo:

1. All'avvio di ogni nodo del cluster, invia una GenerateDataKeyrichiesta a AWS KMS, specificando l'ID della KMS chiave scelta quando è stata abilitata 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 non crittografato) e l'altra copia è crittografata sotto la chiave. KMS
3. Il nodo utilizza una versione con codifica Base64 della chiave dati in testo semplice 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 con una richiesta Decrypt. AWS KMS
5. AWS KMS decrittografa la chiave dati crittografata utilizzando la stessa KMS chiave utilizzata per crittografarla, quindi invia la chiave di dati decrittografata (testo semplice) al nodo.

6. Il nodo utilizza la versione con codifica base64 della chiave dati in testo semplice 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 KMS chiave specifica.

La sezione seguente spiega il contesto di crittografia utilizzato in ogni scenario di EMR crittografia Amazon che utilizza una KMS chiave.

Contesto di EMRFS crittografia per la crittografia con SSE - KMS

Con SSE -KMS, il EMR cluster Amazon invia i dati ad Amazon S3, quindi Amazon S3 utilizza KMS una chiave 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 GenerateDataKeyand Decrypt a cui viene inviata. AWS KMS L'esempio seguente mostra una JSON rappresentazione del contesto di crittografia utilizzato da Amazon S3.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Contesto di crittografia per la EMRFS crittografia con CSE - KMS

Con CSE -KMS, il EMR cluster Amazon utilizza una KMS chiave per crittografare i dati prima di inviarli ad Amazon S3 per l'archiviazione. In questo caso, il cluster utilizza l'Amazon Resource Name (ARN) della KMS chiave come contesto di crittografia per ogni richiesta GenerateDataKeyand Decrypt a cui invia. AWS KMS L'esempio seguente mostra una JSON rappresentazione del contesto di crittografia utilizzato dal cluster.

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Contesto di crittografia per la crittografia del disco locale con LUKS

Quando un EMR cluster Amazon utilizza la crittografia del disco locale conLUKS, i nodi del cluster non specificano il contesto di crittografia con le richieste GenerateDataKeye 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 an AWS KMS key come chiave principale.

Le chiavi di crittografia dei dati crittografano i blocchi di dati nel cluster. A ogni blocco di dati viene assegnata una chiave -256 generata casualmenteAES. 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 -256 generata casualmente. AES 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 di crittografia simmetriche. KMS Non è possibile utilizzare una KMS chiave asimmetrica in un flusso di lavoro di crittografia Amazon Redshift. Per informazioni su come determinare se una KMS chiave 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) AWS KMS utilizzati per verificare l'integrità dei dati. 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 (KMSchiave). 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 KMS chiavi.

- Documentazione di riferimento dei tipi di chiave. Elenca il tipo di KMS chiave che supporta ogni AWS KMS API operazione.

  Per trovare: Posso abilitare e disabilitare una KMS chiave di RSA firma?

- Tabella dello stato delle chiavi. Mostra come lo stato della KMS chiave influenzi il suo utilizzo nelle AWS KMS API operazioni.

  Per trovare: è possibile modificare l'alias di una KMS chiave in attesa di eliminazione?

- AWS KMS APIriferimento alle autorizzazioni. Fornisce informazioni sulle autorizzazioni richieste per ogni AWS KMS API operazione.

  Per trovare: Posso eseguire GetKeyPolicyl'esecuzione con una chiave in un altro AWS account? Posso consentire `kms:Decrypt` l'autorizzazione in una IAM politica?

  - 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 KMS chiavi.

  Per scoprire: quanto costa usare le chiavi asimmetriche?

- AWS KMS richiedere quote. Elenca le quote al secondo per le AWS KMS API richieste in ogni account e regione.

  Per scoprire: quante richieste Decrypt possono essere eseguite al secondo? Quante richieste di decrittografia posso eseguire sulle KMS chiavi nel mio archivio di chiavi personalizzate?

- AWS KMS quote di risorse. Elenca le quote delle risorse AWS KMS .

  Per trovare: quante KMS chiavi posso avere in ogni regione del mio account? Quanti alias posso avere su ogni KMS chiave?

- AWS servizi integrati con AWS KMS. Elenca i AWS servizi che utilizzano KMS le chiavi per proteggere le risorse che creano, archiviano e gestiscono.

Per scoprire: Amazon Connect utilizza KMS le chiavi per proteggere le mie risorse Connect?

Stati chiave delle AWS KMS chiavi

An ha AWS KMS key sempre uno stato chiave. Le operazioni sulla KMS chiave e sul relativo ambiente possono modificare lo stato della chiave, in modo transitorio o finché un'altra operazione non ne modifica lo stato della chiave.

La tabella in questa sezione mostra come gli stati chiave influiscono sulle chiamate alle AWS KMS API operazioni. A causa dello stato della chiave, ci si aspetta che un'operazione su una KMS chiave abbia successo (#), fallisca (X) o abbia successo solo in determinate condizioni (?). Il risultato è spesso diverso per KMS le chiavi con materiale chiave importato.

Questa tabella include solo le API operazioni che utilizzano una KMS chiave esistente. Le altre operazioni, come CreateKeye ListKeys, vengono omesse.

Argomenti

- Stati e tipi di KMS chiave chiave
- Tabella dello stato delle chiavi

Stati e tipi di KMS chiave chiave

Il tipo di KMS chiave determina gli stati chiave che può avere.

- Tutte KMS le chiavi possono trovarsi negli `PendingDeletion` stati `Enabled``Disabled`, e.
- La maggior parte delle KMS chiavi viene creata nello `Enabled` stato. Le chiavi KMS con il materiale chiave importato vengono create nello stato `PendingImport`.
- Lo `PendingImport` stato si applica solo alle KMS chiavi con materiale chiave importato.
- Lo `Unavailable` stato si applica solo a una KMS chiave in un archivio di chiavi personalizzato. Una KMS chiave in un archivio AWS CloudHSM chiavi `Unavailable` si verifica quando l'archivio chiavi personalizzato viene disconnesso intenzionalmente dal relativo AWS CloudHSM cluster. Una KMS chiave in un archivio chiavi esterno si verifica `Unavailable` quando l'archivio chiavi personalizzato viene intenzionalmente disconnesso dal proxy dell'archivio chiavi esterno. È possibile visualizzare e gestire le KMS chiavi non disponibili, ma non è possibile utilizzarle nelle operazioni crittografiche.

Lo stato della KMS chiave in un archivio chiavi personalizzato non è influenzato dalle modifiche apportate alla relativa chiave di supporto. Una KMS chiave in un archivio AWS CloudHSM chiavi non è influenzata dalle modifiche al materiale chiave associato nel AWS CloudHSM cluster. Una KMS chiave in un archivio di chiavi esterno non è influenzata dalle modifiche apportate alla relativa chiave esterna in un gestore di chiavi esterno. Se la chiave di supporto è disabilitata o eliminata, lo stato della KMS chiave non cambia, ma le operazioni di crittografia che utilizzano la KMS chiave falliscono.

- 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'ReplicateKeyoperazione. 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'UpdatePrimaryRegionoperazione. 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

La tabella seguente mostra come lo stato della KMS chiave influenzi AWS KMS le operazioni.

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
CancelKeyDeletion	 [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							
DeleteImportedKeyMaterial	 [9]	 [9]	 [9]	 (nessun effetto)	N/D	 [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
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]
EnableKeyRotation	? [7]	⊗ [1] o [7]	⊗ [3] o [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]
Crittografa	✓	⊗ [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
GenerateDataKey	✓	⊗ [1]	⊗ [2] o [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyPair	✓	⊗ [1]	⊗ [2] o [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	⊗ [1]	⊗ [2] o [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	⊗ [1]	⊗ [2] o [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateMac	✓	⊗ [1]	⊗ [2] o [3]	N/D	N/D	⊗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
GetKeyRotationStatus	 [7]	 [7]	 [7]	 [6]	 [7]	 [7]	 [7]
GetParametersForImport	 [9]	 [9]	 [8] o [9]		 [9]	 [14]	 [15]
GetPublicKey			 [2] o [3]	N/D	N/D	 [14]	
ImportKeyMaterial	 [9]	 [9]	 [8] o [9]		 [9]	 [14]	
ListAliases							
ListGrants							
ListKeyPolicies							

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
ListKeyRotations	 [7]	 [7]	 [7]	 [6]	 [7]	 [7]	 [7]
ListResourceTags							
PutKeyPolicy							
ReEncrypt		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
ReplicateKey		 [1]	 [2] o [3]	 [5]	N/D	 [14]	 [15]
RetireGrant							
RevokeGrant							

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
RotateKeyOnDemand	[7]	[1] o [7]	[3] o [7]	[6]	[7]	[14]	[7]
ScheduleKeyDeletion	✓	✓	[3]	✓	✓	✓	[15]
Sign	✓	[1]	[2] o [3]	N/D	N/D	[14]	✓
TagResource	✓	✓	[3]	✓	✓	✓	✓
UntagResource	✓	✓	[3]	✓	✓	✓	✓
UpdateAlias	✓	✓	[10]	✓	✓	✓	✓

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
UpdateKeyDescription	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊗ [1]	⊗ [2] o [3]	⊗ [5]	N/D	⊗ [14]	✓
Verifica	✓	⊗ [1]	⊗ [2] o [3]	N/D	N/D	⊗ [14]	✓
VerifyMac	✓	⊗ [1]	⊗ [2] o [3]	N/D	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.`
- [6] `UnsupportedOperationException: <key ARN> origin is EXTERNAL which is not valid for this operation.`
- [7] Se la KMS chiave ha importato materiale chiave o si trova in un archivio di chiavi personalizzato:`UnsupportedOperationException`.
- [8] Se la KMS chiave ha importato materiale chiave: `KMSInvalidStateException`
- [9] Se la KMS chiave non può o non ha materiale chiave importato:`UnsupportedOperationException`.
- [10] Se la KMS chiave sorgente è in attesa di eliminazione, il comando ha esito positivo. Se la KMS chiave di destinazione è in attesa di eliminazione, il comando ha esito negativo e restituisce un errore: `KMSInvalidStateException : <key ARN> is pending deletion.`
- [11] Non `KMSInvalidStateException: <key ARN> is unavailable.` è possibile eseguire questa operazione su una KMS chiave non disponibile.
- [12] L'operazione ha esito positivo, ma lo stato della KMS chiave non cambia finché non diventa disponibile.
- [13] Sebbene una KMS chiave in un archivio chiavi personalizzato sia in attesa di eliminazione, lo stato della chiave rimane invariato `PendingDeletion` anche se la KMS chiave non è più disponibile. Ciò consente di annullare l'eliminazione della KMS chiave in qualsiasi momento durante il periodo di attesa.
- [14] `KMSInvalidStateException: <key ARN> is creating.` AWS KMS genera questa eccezione durante la replica di 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 KMS chiavi. Ad esempio, è possibile utilizzare solo chiavi di crittografia simmetriche per generare KMS chiavi dati simmetriche e coppie di chiavi dati asimmetriche. Inoltre, l'importazione di materiale chiave e la rotazione automatica delle chiavi sono supportate solo per le chiavi di crittografia simmetriche ed è possibile creare solo KMS chiavi di crittografia simmetriche in un archivio di chiavi personalizzato. KMS

Questo riferimento include due tabelle.

- La tabella dei tipi di chiave elenca le AWS KMS operazioni valide per le chiavi di crittografia simmetriche, le chiavi asimmetriche e KMS le chiavi. KMS HMAC KMS
- La tabella delle caratteristiche speciali elenca le AWS KMS operazioni valide per le chiavi multiregionali, KMS le chiavi con materiale chiave importato e KMS KMS le chiavi negli archivi di chiavi personalizzati.

Tabella dei tipi di chiave

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) 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
DeleteAlias	Sì	Sì	Sì	Sì	Sì
DeleteImportedKeyMaterial Valido solo su KMS chiavi con materiale chiave importato	Sì	Sì	Sì	Sì	Sì

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
(`Origini`). `EXTERNAL`					
DeriveSharedSecret	No	No	No	No	Sì
DescribeKey	Sì	Sì	Sì	Sì	Sì
DisableKey	Sì	Sì	Sì	Sì	Sì
DisableKeyRotation	Sì Valido solo su KMS AWS KMS chiavi con (`Origini` materiale chiave.	No	No	No	No
EnableKey	Sì	Sì	Sì	Sì	Sì

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPTDECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
EnableKeyRotation	Sì Valido solo su KMS AWS KMS chiavi con (`0rigini/` materiale chiave.	No	No	No	No
Encrypt	Sì	No	Sì	No	No
GenerateDataKey	Sì	No	No	No	No
GenerateDataKeyPair Genera una coppia di chiavi dati asimmetriche protetta da una chiave di crittografia simmetrica. KMS	Sì Non valido per le chiavi negli archivi di KMS chiavi personalizzati.	No	No	No	No

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPTDECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
GenerateDataKeyPairWithoutPlaintext Genera una coppia di chiavi dati asimmetriche protetta da una chiave di crittografia simmetrica. KMS	Sì Non valido per le chiavi negli archivi di KMS chiavi personalizzati.	No	No	No	No
GenerateDataKeyWithoutPlaintext	Sì	No	No	No	No
GenerateMac	No	Sì	No	No	No
GetKeyPolicy	Sì	Sì	Sì	Sì	Sì
GetKeyRotationStatus	Sì	Sì (KeyRotaonEnablsarà sempre false).	Sì (KeyRotaonEnablsarà sempre false).	Sì (KeyRotationEnabledsarà sempre false).	Sì (KeyRotationEnabledsarà sempre false).

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
GetParametersForImport Valido solo sulle KMS chiavi con materiale chiave importato. `Origin` `EXTERNAL`	Sì	Sì	Sì	Sì	Sì
GetPublicKey	No	No	Sì	Sì	Sì
ImportKeyMaterial Valido solo sulle KMS chiavi con materiale chiave importato. `Origin` `EXTERNAL`	Sì	Sì	Sì	Sì	Sì
ListAliases	Sì	Sì	Sì	Sì	Sì
ListGrants	Sì	Sì	Sì	Sì	Sì
ListKeyPolicies	Sì	Sì	Sì	Sì	Sì

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
ListKeyRotations	Sì	Sì (Il `Rotatio` campo sarà sempre nullo o vuoto.)	Sì (Il `Rotatio` 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 - Valido solo su chiavi multi-Regione	Sì	Sì	Sì	Sì	Sì
RetireGrant	Sì	Sì	Sì	Sì	Sì
RevokeGrant	Sì	Sì	Sì	Sì	Sì

AWS KMS APIoperazione	chiavi di crittografia simmetriche KMS	HMACKM iavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
RotateKeyOnDemand	Sì Valido solo su KMS chiavi con materiale AWS KMS chiave (`Originis`	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 APIoperazione	chiavi di crittografia simmetriche KMS	HMACKMiavi	KMStasti asimmetrici (_) ENCRYPT DECRYPT	Tasti asimmetrici KMS (_) SIGN VERIFY	Tasti asimmetrici KMS (_) KEY AGREEMENT
UpdateAlias La KMS chiave corrente e la nuova KMS chiave devono essere dello stesso tipo (entrambe simmetriche o entrambe asimmetriche o entrambeHMAC) e devono avere lo stesso utilizzo della chiave.	Sì	Sì	Sì	Sì	Sì
UpdateKey Description	Sì	Sì	Sì	Sì	Sì
UpdateReplicaRegion - Valido solo su chiavi multi-Regione	Sì	Sì	Sì	Sì	Sì
Verify	No	No	No	Sì	No
VerifyMac	No	Sì	No	No	No

Tabella delle caratteristiche speciali

Questa tabella mostra le AWS KMS API operazioni supportate su ogni tipo di chiave speciale.

Durante la lettura di questa tabella, tieni a mente le interazioni seguenti:

- Chiavi multi-regione:
  - Le chiavi multiregionali possono essere chiavi di crittografia simmetriche, KMS chiavi asimmetriche, chiavi e KMS chiavi con materiale chiave HMAC KMS importato. KMS
  - Non è possibile creare chiavi multi-regione in un archivio delle chiavi personalizzate.
- Materiale della chiave importato
  - È possibile importare materiale chiave per chiavi di crittografia KMS simmetriche, chiavi asimmetriche e chiavi. KMS HMAC KMS
  - 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 KMS le chiavi con materiale chiave importato.
- store delle chiavi personalizzate
  - Gli archivi di chiavi personalizzati supportano solo chiavi di crittografia KMS simmetriche.
  - Le operazioni simmetriche su coppie di chiavi asimmetriche (`GenerateDataKeyPair`,`GenerateDataKeyPairWithoutPlaintext`) non sono supportate sulle chiavi negli archivi di chiavi personalizzati. KMS
  - La rotazione automatica delle chiavi (`EnableKeyRotation`,`DisableKeyRotation`) non è supportata sulle KMS chiavi negli archivi di chiavi personalizzati.
  - 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 APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey È possibile utilizzare `CreateKey` per creare una chiave primaria multiregionale, una KMS chiave con materiale chiave importato o una KMS chiave in un archivio di chiavi personalizzato. Per creare una chiave di replica multi-regione, utilizza `ReplicateKey` .	✓	✓	✓
Decrypt	✓ Valido solo quando `KeyUsage` è `ENCRYPT_DECRYPT`	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ Valido solo per le chiavi con	✓	⊗

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
	materiale della chiave importato (`Origin` è `EXTERNAL`)		
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓
DisableKeyRotation	✓ Valido solo su chiavi di crittografia simmetriche con uno o più materiali AWS KMS chiave. `Origin` `AWS_KMS`	⊗	⊗
EnableKey	✓ Valido solo su chiavi di crittografia simmetriche KMS	✓	✓

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
EnableKeyRotation	✓ Valido solo su chiavi di crittografia simmetriche con uno o più materiali AWS KMS chiave. `Origin` `AWS_KMS`	⊗	⊗
Encrypt	✓ Valido solo quando `KeyUsage` è `ENCRYPT_DECRYPT`	✓	✓
GenerateDataKey	✓ Valido solo su chiavi di crittografia simmetriche KMS	✓	✓
GenerateDataKeyPair	✓ Valido solo su chiavi di crittografia simmetriche KMS	✓	⊗

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
GenerateDataKeyPairWithoutPlaintext	✓ Valido solo su chiavi di crittografia simmetriche KMS	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ Valido solo su chiavi di crittografia simmetriche KMS	✓	✓
GenerateMac Valido solo per le chiavi HMAC KMS	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` sarà sempre `false`).	⊗

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
GetParametersForImport	✓ Valido solo per le chiavi con materiale della chiave importato (`Origin` è `EXTERNAL`).	✓	⊗
GetPublicKey Valido solo per chiavi asimmetriche KMS.	✓	✓	⊗
ImportKeyMaterial	✓ Valido solo per le chiavi con materiale della chiave importato (`Origin` è `EXTERNAL`).	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓
ListResourceTags	✓	✓	✓

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Valido solo quando `KeyUsage` è `ENCRYPT_DECRYPT`	✓	✓
ReplicateKey	✓ Valido solo su chiavi primarie multi-regione.	✓ Valido solo su chiavi primarie multi-regione.	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓
Sign Valido solo quando `KeyUsage` è `SIGN_VERIFY` .	✓	✓	⊗
TagResource	✓	✓	✓

AWS KMS APIoperazione	Chiavi multi-regione	Materiale della chiave importato	KMSchiavi in un archivio di chiavi personalizzato
UntagResource	✓	✓	✓
UpdateAlias - La KMS chiave corrente e la nuova KMS chiave devono essere dello stesso tipo (entrambe simmetriche o entrambe asimmetriche o entrambeHMAC) 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 HMACKMSValido solo sulle chiavi	✓	✓	⊗

Riferimento alle specifiche chiave

Quando si crea una chiave asimmetrica o una KMS chiave, si seleziona HMAC KMS la relativa specifica chiave. La specifica della chiave, che è una proprietà di every AWS KMS key, rappresenta la configurazione crittografica della chiave. KMS Le specifiche della chiave vengono scelte al

momento della creazione della KMS chiave e non possono essere modificate. Se hai selezionato la specifica chiave sbagliata, elimina la KMS chiave e creane una nuova.

> **Note**
>
> La specifica chiave di una KMS chiave era nota come «specifica chiave master del cliente». Il `CustomerMasterKeySpec` parametro dell'CreateKeyoperazione è obsoleto. Utilizza invece il parametro `KeySpec`. La risposta delle DescribeKeyoperazioni `CreateKey` and include un `CustomerMasterKeySpec` membro `KeySpec` and con lo stesso valore.

Le specifiche della chiave determinano se la KMS chiave è simmetrica o asimmetrica, il tipo di materiale chiave contenuto nella chiave e gli algoritmi di crittografia, gli algoritmi di firma o gli algoritmi del codice di autenticazione dei messaggi (MAC) supportati dalla KMS chiave. AWS KMS KMS La specifica della chiave scelta è in genere determinata dal caso d'uso e dai requisiti normativi. Tuttavia, le operazioni crittografiche su KMS chiavi con specifiche chiave diverse hanno prezzi diversi e sono soggette a quote diverse. 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 mandanti possono utilizzare durante la creazione delle KMS chiavi, usa la chiave kms: condition. KeySpec Puoi anche usare la chiave `kms:KeySpec` condition per consentire ai principali di richiamare AWS KMS operazioni solo su KMS chiavi con una particolare specifica chiave. Ad esempio, è possibile negare l'autorizzazione a pianificare l'eliminazione di qualsiasi KMS chiave con una `RSA_4096` specifica chiave.

AWS KMS supporta le seguenti specifiche chiave per KMS le chiavi:

Specifica della chiave crittografica simmetrica (impostazione predefinita)

- SYMMETRIC_DEFAULT

RSAspecifiche chiave (crittografia e decrittografia -oppure- firma e verifica)

- RSA_2048
- RSA_3072
- RSA_4096

Specifiche della chiave basata su curva ellittica

- Coppie di chiavi a curva ellittica asimmetriche NIST consigliate (firma e verifica, o derivazione di segreti condivisi)

    - ECC_ NIST _P256 (secp256r1)
    - ECCNIST_ P384 (secp384r1)
    - ECCNIST_ P521 (secp521r1)
- Altre coppie di chiavi asimmetriche basate su curva ellittica (firma e verifica)
    - ECC_ SECG _P256K1 (secp256k1), comunemente usato per le criptovalute.

SM2specifica chiave (crittografia e decrittografia -oppure - firma e verifica -o- derivazione di segreti condivisi)

- SM2(Solo regioni della Cina)

HMACspecifiche chiave

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

SYMMETRIC_ specifiche DEFAULT chiave

La specifica chiave predefinita, SYMMETRIC _DEFAULT, è la specifica chiave per le chiavi di crittografia simmetriche. KMS 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'CreateKeyoperazione, se non si specifica un valore, viene selezionato _. `KeySpec` SYMMETRIC DEFAULT Se non avete motivo di utilizzare una specifica chiave diversa, SYMMETRIC _ DEFAULT è una buona scelta.

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 dati autenticati aggiuntivi (AAD), come un contesto di crittografia, e GCM fornisce un ulteriore controllo di integrità sul testo cifrato.

I dati crittografati con AES -256- GCM sono protetti ora e in futuro. I crittografi considerano questo algoritmo resistente alla quantistica. Futuri teorici, attacchi informatici quantistici su larga scala su testi cifrati creati a 256 bitAES: GCM le chiavi riducono la sicurezza effettiva 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 cinesi, dove SYMMETRIC _ DEFAULT rappresenta una chiave simmetrica a 128 bit che utilizza la crittografia. SM4 È possibile creare una SM4 chiave a 128 bit solo all'interno delle regioni cinesi. Non è possibile creare una GCM KMS chiave a 256 bit nelle AES regioni cinesi.

È possibile utilizzare una KMS chiave di crittografia simmetrica per crittografare, AWS KMS decrittografare e ricrittografare i dati e per proteggere le chiavi e le coppie di chiavi di dati generate. AWS i servizi integrati con AWS KMS utilizzano chiavi di crittografia simmetriche per crittografare i dati inattivi. KMS È possibile importare le proprie chiavi in una chiave di crittografia simmetrica e creare KMS chiavi di crittografia simmetriche in archivi di chiavi personalizzati. KMS Per una tabella che confronta le operazioni che è possibile eseguire su chiavi simmetriche e asimmetriche, vedere Confronto tra chiavi simmetriche e asimmetricheKMS. KMS

È possibile utilizzare una chiave di crittografia simmetrica per crittografare, decrittografare e AWS KMS ricrittografare i dati e generare KMS chiavi di dati e coppie di chiavi di dati. È possibile creare chiavi di crittografia simmetriche multiregionali, importare il proprio materiale chiave in una chiave di crittografia simmetrica e creare KMS chiavi di crittografia KMS simmetriche in archivi di chiavi personalizzati. KMS Per una tabella che confronta le operazioni che è possibile eseguire su KMS chiavi di diversi tipi, vedere. Documentazione di riferimento dei tipi di chiave

RSAspecifiche principali

Quando si utilizza una specifica RSA chiave, AWS KMS crea una chiave asimmetrica con una coppia di KMS 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 utilizzi la chiave pubblica di una KMS chiave che è stata eliminata da AWS KMS, la chiave pubblica di una KMS chiave configurata per la firma e la verifica o un algoritmo di crittografia che non è supportato dalla KMS chiave, i dati sono irrecuperabili.

In AWS KMS, puoi utilizzare chiavi asimmetriche con coppie di KMS chiavi per la crittografia e RSA la decrittografia o la firma e la verifica, ma non entrambe. Questa proprietà, nota come utilizzo della chiave, viene determinata separatamente dalla specifica della chiave; tuttavia è preferibile che tu la definisca prima di selezionare la specifica.

AWS KMS supporta le seguenti specifiche RSA chiave per la crittografia e la decrittografia o la firma e la verifica:

- RSA_2048
- RSA_3072
- RSA_4096

RSAle specifiche chiave differiscono in base alla lunghezza della RSA chiave in bit. Le specifiche RSA chiave scelte potrebbero essere determinate dagli standard di sicurezza o dai requisiti dell'attività. In generale, usa la chiave più grande che ritieni pratica e conveniente per la tua attività. Le operazioni crittografiche su KMS chiavi con specifiche RSA chiave diverse hanno prezzi diversi. 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.

RSAspecifiche chiave per la crittografia e la decrittografia

Quando si utilizza una KMS chiave RSA asimmetrica per la crittografia e la decrittografia, si esegue la crittografia con la chiave pubblica e la decrittografia con la chiave privata. Quando richiami l'`Encrypt`operazione AWS KMS per una RSA KMS chiave, AWS KMS utilizza la chiave pubblica nella coppia di RSA chiavi e l'algoritmo di crittografia specificato per crittografare i dati. Per decrittografare il testo cifrato, richiamate l'`Decrypt`operazione e specificate la stessa chiave e lo stesso KMS algoritmo di crittografia. AWS KMS utilizza quindi la chiave privata nella coppia di RSA chiavi 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 RSA KMS chiavi. Per decrittografare il testo cifrato, chiamate la `Decrypt` funzione con la stessa KMS chiave e lo stesso algoritmo di crittografia.

AWS KMS supporta due algoritmi di crittografia per KMS chiavi con specifiche chiave. RSA Questi algoritmi, definiti in PKCS#1 v2.2, differiscono nella funzione hash che utilizzano internamente. In AWS KMS, gli OAEP algoritmi RSAES _ utilizzano sempre la stessa funzione di hash sia per scopi di hashing che per la funzione di generazione della maschera (). 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 chiave RSA

Algoritmo di crittografia	Descrizione dell'algoritmo
RSAES_ OAEP _ _1 SHA	PKCS#1 v2.2, Sezione 7.1. RSAcrittografia con OAEP Padding che utilizza SHA -1 sia per l'hash che per la funzione di generazione della MGF1 maschera insieme a un'etichetta vuota.
RSAES_ _ _256 OAEP SHA	PKCS#1, Sezione 7.1. RSAcrittografia con OAEP Padding che utilizza SHA -256 sia per l'hash che per la funzione di generazione della MGF1 maschera insieme a un'etichetta vuota.

Non è possibile configurare una KMS chiave per utilizzare un particolare algoritmo di crittografia. Tuttavia, è possibile 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 KMS chiave, visualizza la configurazione crittografica della KMS chiave nella AWS KMS console o usa l'operazione. DescribeKey AWS KMS fornisce inoltre le specifiche della chiave e gli algoritmi di crittografia quando si scarica la chiave pubblica, nella AWS KMS console o utilizzando l'operazione. GetPublicKey

È possibile scegliere una specifica RSA chiave in base alla lunghezza dei dati di testo in chiaro che è possibile 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 fare un confronto, è possibile utilizzare una KMS chiave di crittografia simmetrica per crittografare fino a 4096 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

	Algoritmo di crittografia	
Specifica della chiave	RSAES_ _ _1 OAEP SHA	RSAES_ OAEP _ _256 SHA
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

RSAspecifiche principali per la firma e la verifica

Quando si utilizza una KMS chiave RSA asimmetrica per la firma e la verifica, si genera la firma per un messaggio con la chiave privata e si verifica la firma con la chiave pubblica.

Quando richiami l'`Sign`operazione AWS KMS per una chiave asimmetrica, AWS KMS utilizza la KMS chiave privata nella coppia di RSA chiavi, nel messaggio e nell'algoritmo di firma specificato per generare una firma. Per verificare la firma, chiamare l'azione Verify. Specificate la firma, più la stessa KMS chiave, lo stesso messaggio e lo stesso algoritmo di firma. AWS KMS utilizza quindi la chiave pubblica nella coppia di RSA chiavi per verificare la firma. Puoi anche scaricare la chiave pubblica e usarla per verificare la firma all'esterno di AWS KMS.

AWS KMS supporta i seguenti algoritmi di firma per tutte le KMS chiavi con una specifica RSA chiave. 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 RSA chiavi, sono preferiti PSS gli algoritmi RSASSA -. Includiamo algoritmi RSASSA - PKCS1 -v1_5 per la compatibilità con le applicazioni esistenti.

Algoritmi di firma supportati per le specifiche chiave RSA

Algoritmo di firma	Descrizione dell'algoritmo
RSASSA_ PSS _ _256 SHA	PKCS#1 v2.2, Sezione 8.1, RSA firma con PSS imbottitura che utilizza SHA -256 sia per il message digest che per la funzione di generazione della MGF1 maschera insieme a un sale a 256 bit

Algoritmo di firma	Descrizione dell'algoritmo
RSASSA_ _ _384 PSS SHA	PKCS#1 v2.2, Sezione 8.1, RSA firma con PSS imbottitura che utilizza SHA -384 sia per il message digest che per la funzione di generazione della MGF1 maschera insieme a un sale a 384 bit
RSASSA_ _ _512 PSS SHA	PKCS#1 v2.2, Sezione 8.1, RSA firma con PSS imbottitura che utilizza SHA -512 sia per il message digest che per la funzione di generazione della MGF1 maschera insieme a un sale a 512 bit
RSASSAPKCS1_ SHA _V1_5_ _256	PKCS#1 v2.2, Sezione 8.2, firma con imbottitura #1v1 .5 e -256 RSA PKCS SHA
RSASSAPKCS1_ SHA _V1_5_ _384	PKCS#1 v2.2, Sezione 8.2, firma con imbottitura #1v1 .5 e -384 RSA PKCS SHA
RSASSAPKCS1_ SHA _V1_5_ _512	PKCS#1 v2.2, Sezione 8.2, firma con imbottitura #1v1 .5 e -512 RSA PKCS SHA

Non è possibile configurare una KMS chiave per utilizzare particolari algoritmi di firma. 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 KMS chiave, visualizza la configurazione crittografica della KMS chiave nella AWS KMS console o utilizzando l'operazione. DescribeKey AWS KMS fornisce inoltre le specifiche della chiave e gli algoritmi di firma quando si scarica 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 elliptic curve (ECC), AWS KMS crea una chiave asimmetrica con KMS una coppia di chiavi per la firma e la verifica o ECC 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

AWS KMS supporta le seguenti specifiche ECC chiave per le chiavi asimmetricheKMS.

- Coppie di chiavi con curva ellittica asimmetrica NIST consigliate (firma e verifica o derivazione di segreti condivisi)
  - ECC_ NIST _P256 (secp256r1)
  - ECCNIST_ P384 (secp384r1)
  - ECCNIST_ P521 (secp521r1)
- Altre coppie di chiavi asimmetriche basate su curva ellittica (firma e verifica)
  - ECC_ SECG _P256K1 (secp256k1), comunemente usato per le criptovalute.

Le specifiche ECC chiave da scegliere potrebbero essere determinate dagli standard di sicurezza o dai requisiti dell'attività. In generale, utilizza la curva con il maggior numero di punti che ritieni pratica e conveniente per la tua attività.

Se stai creando una chiave asimmetrica per ricavare segreti condivisi, usa una delle specifiche KMS chiave consigliate per la NIST curva ellittica. L'unico algoritmo di accordo chiave 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 asimmetrica da usare con le criptovalute, usa la specifica KMS chiave _ _P256K1. ECC SECG Puoi utilizzare anche questa specifica della chiave per altri scopi, ma è necessaria per il Bitcoin e le altre criptovalute.

KMSle chiavi con specifiche ECC chiave diverse hanno prezzi diversi e sono soggette a quote di richiesta diverse. Per informazioni sui AWS KMS prezzi, consulta la sezione AWS Key Management Service Prezzi. Per informazioni sulle quote di richieste, consulta Quote di richieste.

La tabella seguente mostra gli algoritmi di firma AWS KMS supportati per ciascuna delle specifiche ECC chiave. Non è possibile configurare una KMS chiave 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 chiave ECC

Specifica della chiave	Algoritmo di firma	Descrizione dell'algoritmo
ECC_ NIST _P256	ECDSA_ _256 SHA	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 digest del messaggio.
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 digest del messaggio.
ECC_ SECG _P256K1	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.

SM2specifiche chiave (solo regioni cinesi)

La specifica SM2 chiave è una specifica chiave a curva ellittica definita all'interno della serie di specifiche GM/T pubblicate dall'Office of State Commercial Cryptography Administration cinese (). OSCCA La specifica SM2 chiave è disponibile solo nelle regioni della Cina. Quando si utilizza la specifica SM2 chiave, AWS KMS crea una chiave asimmetrica con una coppia di KMS 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 KMS chiave può essere utilizzata solo una chiave. È possibile utilizzare una SM2 KMS chiave per la firma e la verifica, la crittografia e la decrittografia o per ricavare segreti condivisi. È necessario specificare l'utilizzo della chiave al momento della creazione della KMS chiave e non è possibile modificarlo dopo la creazione della chiave.

Se stai creando una KMS chiave asimmetrica per derivare segreti condivisi, usa la specifica 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:

- SM2PKEalgoritmo di crittografia

  SM2PKEè un algoritmo di crittografia basato su curve ellittiche definito OSCCA in GM/T 0003.4-2012.

- SM2DSAalgoritmo di firma

  SM2DSAè un algoritmo di firma basato su curve ellittiche definito OSCCA in GM/T 0003.2-2012. SM2DSArichiede un ID distintivo sottoposto a hash con l'algoritmo di SM3 hashing e quindi combinato con il messaggio, o message digest, a cui è stato passato. AWS KMS Questo valore concatenato viene quindi sottoposto a hash e firmato da. AWS KMS

Specifiche chiave delle chiavi HMAC KMS

AWS KMS supporta HMAC chiavi 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'MACalgoritmo utilizzato e le operazioni. GenerateMacVerifyMac In generale, le chiavi più lunghe sono più sicure. Usa la chiave più lunga funzionale per il tuo caso d'uso.

HMACspecifiche chiave	MACalgoritmo
HMAC_224	HMAC_ _224 SHA
HMAC_256	HMAC_ _256 SHA
HMAC_384	HMAC_ _384 SHA
HMAC_512	HMAC_ _512 SHA

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 di crittografia simmetriche, KMS le chiavi asimmetriche e le KMS chiavi, consulta la. HMAC KMS 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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
ConnectCustomKeyStore `kms:ConnectCustomKeyStore`	IAMpolitica	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
CreateAlias `kms:CreateAlias` Per utilizzare questa operazione, il chiamante necessita dell'auto rizzazione `kms:CreateAlias` su due risorse: • L'alias (in una IAM politica) • La KMS chiave (in una politica chiave) Per informazioni dettaglia te, consultare Controllo dell'accesso agli alias.	Policy IAM (per l'alias)	No	Alias	Nessuna (in caso di controllo dell'accesso all'alias)
	Politica chiave (per la KMS chiave)	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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`	IAMpolitica	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	KMSchiave	Condizioni per il contesto di crittografia: kms:EncryptionContext: chiave contestuale km: EncryptionContextKeys Condizioni di concessione: km: GrantConstraintType km: GranteePrincipal km: GrantIsFor AWSResource km: GrantOperations km: RetiringPrincipal Condizioni per le operazioni KMS chiave: 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
CreateKey `kms:CreateKey`	IAMpolitica	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	KMSchiave	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 KMS chiave: 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 `kms:DeleteAlias`	Policy IAM (per l'alias)	No	Alias	Nessuna (in caso di controllo dell'accesso all'alias)
Per utilizzare questa operazione, il chiamante necessita dell'auto rizzazione `kms:DeleteAlias` su due risorse: • L'alias (in una IAM politica) • La KMS chiave (in una politica chiave) Per informazioni dettaglia te, consultare Controllo dell'accesso agli alias.	Politica chiave (per la KMS chiave)	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:DeleteCustomKeyStore`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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
DedriveSharedSecret `kms:DeriveSharedSecret`	Policy della chiave	Sì	KMSchiave	Condizioni per le operazioni KMS chiave: 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`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:DisableKeyRotation`	Policy della chiave	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:DisconnectCustomKeyStore`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	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 KMS chiave: 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	KMSchiave	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 KMS chiave: 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 `kms:GenerateDataKeyPair`	Policy della chiave	Sì	KMSchiave Genera una coppia di chiavi dati asimmetri che protetta da una chiave di crittografia simmetrica. KMS	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 KMS chiave: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				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 `kms:GenerateDataKeyPairWithoutPlaintext`	Policy della chiave	Sì	KMSchiave Genera una coppia di chiavi dati asimmetriche protetta da una chiave di crittografia simmetrica. KMS	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 KMS chiave: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				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 `kms:GenerateDataKeyWithoutPlaintext`	Policy della chiave	Sì	KMSchiave	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 KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:GetParametersForImport`	Policy della chiave	No	KMSchiave	km: WrappingAlgorithm km: WrappingKeySpec Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:ImportKeyMaterial`	Policy della chiave	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:ListAliases`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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: GrantIsFor AWSResource

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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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`	IAMpolitica	No	*	Nessuno

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListResourceTags `kms:ListResourceTags`	Policy della chiave	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:ListRetirableGrants`	IAMpolitica	L'entità principale specificata 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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
 	
`kms:ReEncryptFrom`
 	
`kms:ReEncryptTo`
 	
Per utilizzare questa operazione, il chiamante necessita dell'auto rizzazione su due tasti: KMS
 	
• `kms:ReEnc ryptFrom` sulla KMS chiave usata per decriptare
• `kms:ReEnc ryptTo` sulla KMS chiave usata per cifrare
	Policy della chiave	Sì	KMSchiave	Condizioni per le operazioni di crittografia
				
				km: EncryptionAlgorithm
				
				km: RequestAlias
				
				Condizioni per il contesto di crittografia:
				
				kms:: chiave contestuale EncryptionContext
				
				km: EncryptionContextK eys
				
				Condizioni per le operazioni KMS chiave:
				
				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: ReEncrypt OnSameKey
ReplicateKey `kms:ReplicateKey` Per utilizzare questa operazione, il chiamante necessita delle seguenti autorizzazioni: • `kms:ReplicateKey` sulla chiave primaria multiregione • `kms:CreateKey` in una IAM politica nella regione di replica	Policy della chiave	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:RetireGrant` L'autorizzazione per ritirare una concessione è determinata principal mente dalla concessione. Una policy da sola non può consentire l'accesso a questa operazione. Per ulteriori informazi oni, consulta Ritirare e revocare le concessioni.	IAMpolitica Questa autorizza zione non è valida in una policy chiave.	Sì	KMSchiave	Condizioni per il contesto di crittografia: kms:EncryptionContext: chiave contestuale km: EncryptionContextK eys Condizioni di concessio ne: km: GrantConstraintType Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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: GrantIsFor AWSResource

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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per la firma e la verifica: km: MessageType km: RequestAlias km: SigningAlgorithm Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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` Per utilizzare questa operazione, il chiamante necessita dell'auto rizzazione `kms:UpdateAlias` su tre risorse: • L'alias • La KMS chiave attualmente associata • La nuova KMS chiave associata Per informazioni dettaglia te, consultare Controllo dell'accesso agli alias.	Policy IAM (per l'alias)	No	Alias	Nessuna (in caso di controllo dell'accesso all'alias)
	Politica chiave (per le KMS chiavi)	No	KMSchiave	Condizioni per le operazioni KMS chiave: 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`	IAMpolitica	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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:UpdatePrimaryRegion` Per utilizzare questa operazione, il chiamante necessita dell'autorizzazione `kms:UpdatePrimaryRegion` 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 `kms:Verify`	Policy della chiave	Sì	KMSchiave	Condizioni per la firma e la verifica: km: MessageType km: RequestAlias km: SigningAlgorithm Condizioni per le operazioni KMS chiave: 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	KMSchiave	Condizioni per le operazioni KMS chiave: 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 AWS KMS API operazione e l'autorizzazione che consente l'operazione. È possibile specificare l'operazione nell'elemento `Action` di un'istruzione di policy.

- Il tipo di criterio indica se l'autorizzazione può essere utilizzata in una politica o IAM in una politica chiave.

  Policy chiave significa che puoi specificare l'autorizzazione nella policy chiave. Quando la politica chiave contiene l'informativa che abilita IAM le politiche, è possibile specificare l'autorizzazione in una IAM politica.

  IAMpolitica significa che è possibile specificare l'autorizzazione solo in una IAM politica.

- 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 titolare di un altro account kms: l'TagResourceautorizzazione a utilizzare una KMS chiave del tuo account, i suoi tentativi di etichettare la KMS chiave nel tuo account falliranno.

- Resources elenca le AWS KMS risorse a cui si applicano le autorizzazioni. AWS KMS supporta due tipi di risorse: una KMS chiave e un alias. In una politica chiave, il valore dell'`Resource`elemento è sempre`*`, il che indica la KMS chiave a cui è associata la politica chiave.

  Utilizzate i seguenti valori per rappresentare una AWS KMS risorsa in una IAM politica.

  KMSchiave

  Quando la risorsa è una KMS chiave, usa la sua chiave ARN. Per assistenza, consulta the section called “Trova l'ID e la chiave della chiave ARN”.

  `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

Se la risorsa è un alias, usa il suo alias ARN. Per assistenza, consulta the section called “Trova il nome e l'alias dell'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 particolare risorsa (chiave o alias), usa un asterisco (). KMS *

In una IAM politica per un' AWS KMS autorizzazione, un asterisco nell'`Resource`elemento indica tutte le AWS KMS risorse (KMSchiavi e alias). È inoltre possibile utilizzare un asterisco nell'`Resource`elemento quando l' AWS KMS autorizzazione non si applica a KMS chiavi o alias particolari. Ad esempio, quando si consente o si `kms:CreateKey` nega `kms:ListKeys` l'autorizzazione, è necessario impostare l'`Resource`elemento su. *

- 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 chiavi 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 FIPS 140-2 moduli di sicurezza hardware certificati (HSMs) 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 di quorum per ogni comando che devono essere soddisfatte per eseguire un comando su. 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 in. HSM Lo stato del dominio viene sincronizzato tra i membri del HSM dominio come token di dominio esportato.

Chiavi di dominio

Tutti gli utenti HSMs di un dominio condividono un set di chiavi di dominio, $\{_{r}DK\}$. Queste chiavi vengono condivise tramite una routine di esportazione dello stato del dominio. Lo stato del dominio esportato può essere importato HSM in qualsiasi paese 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 KMS chiavi esistenti crittografate con la chiave di dominio in uscita vengono ricrittografate 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 relative al quorum per ogni comando che devono essere soddisfatte per eseguire un comando su un membro del HSM dominio.

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 prodotta da un membroHSM, necessariamente, del dominio che ha esportato lo stato del dominio.

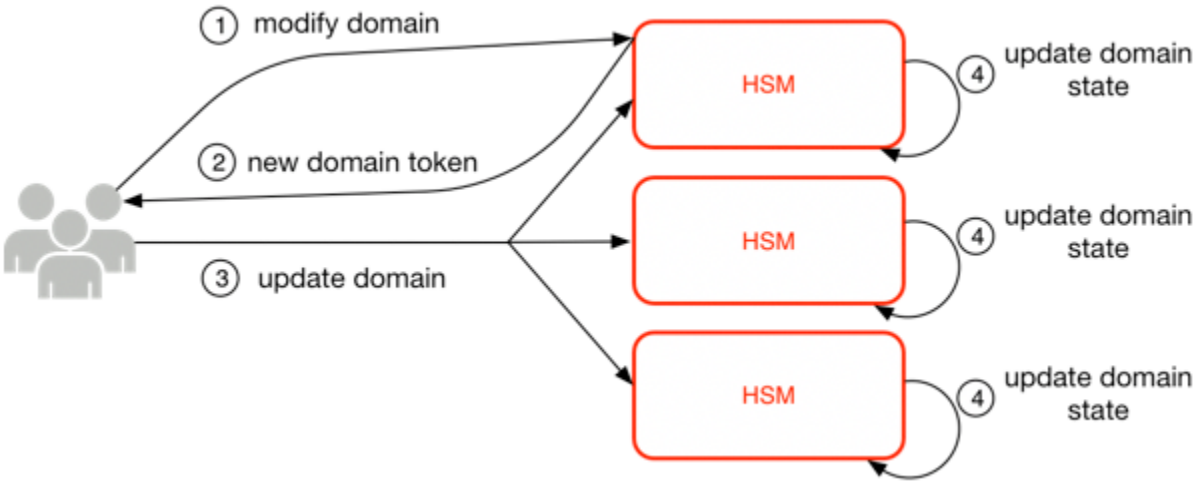
Il token di dominio esportato costituisce la base dell'attendibilità per le entità che operano all'interno del dominio.

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 del dominio, la modifica delle regole del quorum per l'esecuzione dei HSM comandi 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.

Nello stato inizializzato e operativo, an HSM contiene un set di chiavi di identità asimmetriche autogenerate, una coppia di chiavi di firma e una coppia di chiavi per l'impostazione delle chiavi. Tramite un processo manuale, un AWS KMS operatore può stabilire un dominio iniziale da creare per la prima volta in una regione. HSM 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 HSM membri definiti nel dominio.

Dopo essere HSM entrato a far parte di un dominio iniziale, è vincolato 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 API operazioni 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 sul 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 su non HSM viene modificato, il che significa che la modifica non viene apportata su. 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.

HSMsnon 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 host di servizio membro del dominio viene utilizzato per distribuire il nuovo stato del dominio HSM a tutti gli utenti del dominio.

L'uscita e l'adesione a un dominio avvengono tramite le funzioni HSM di gestione. La modifica dello stato del dominio avviene tramite le funzioni di gestione del dominio.

Abbandona dominio

HSMFa sì che an lasci un dominio, eliminando tutti i resti e le chiavi di quel dominio dalla memoria.

Unisci dominio

HSMFa sì che un utente 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

Fa sì che un nuovo dominio venga creato su unHSM. 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 utente HSM dall'elenco degli autorizzati HSMs nel dominio.

Modifica regole

Modifica l'insieme di regole di quorum necessarie per eseguire 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 inSessioni autenticate: un metodo di richiesta firmato dal quorum e una sessione autenticata che utilizza un protocollo -service host. 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 che vengono eseguiti durante le sessioni autenticate aiutano a garantire che solo gli operatori di servizio autorizzati possano eseguire operazioni che coinvolgono le 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 (revisione 2). Questo schema ha un iniziatore con una chiave di firma statica. L'iniziatore genera e firma una chiave Diffie-Hellman () a curva ellittica effimera, destinata a un destinatario con una chiave di accordo statica. ECDH ECDH Questo metodo utilizza una chiave temporanea e due chiavi statiche utilizzando. ECDH Questa è la derivazione dell'etichetta C (1, 2, DH). ECC Questo metodo viene talvolta chiamato one-pass. ECDH

Il secondo metodo di determinazione delle chiavi è C (2, 2ECC, DH). In questo schema, entrambe le parti dispongono di una chiave di firma statica e generano, firmano e scambiano una chiave ECDH effimera. Questo metodo utilizza due chiavi statiche e due chiavi temporanee, ognuna delle quali utilizza. ECDH Questa è la derivazione dell'etichetta C (2, 2,, DH). ECC Questo metodo viene talvolta chiamato ECDH effimero o. ECDHE Tutte le ECDH chiavi vengono generate sulla curva secp384r1 (-P384). NIST

HSMlimite di sicurezza

Il limite di sicurezza interno di AWS KMS è il. HSM HSMHa un'interfaccia proprietaria e non ha altre interfacce fisiche attive nel suo stato operativo. Durante l'inizializzazione, a un operatore vengono

fornite le chiavi crittografiche necessarie per stabilire il suo ruolo nel dominio. HSM I materiali crittografici sensibili di HSM vengono archiviati solo nella memoria volatile e cancellati quando HSM esce dallo stato operativo, inclusi arresti o ripristini intenzionali o non intenzionali.

Le HSM API operazioni vengono autenticate mediante singoli comandi o tramite una sessione riservata con autenticazione reciproca stabilita da un host del 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 insieme 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}. $_{t\ t}$ Affinché un comando soddisfi la regola del quorum, il set di dati del comando 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 con doppia firma viene inviato a un utente per l'esecuzione. HSM HSMEsegue quanto segue:

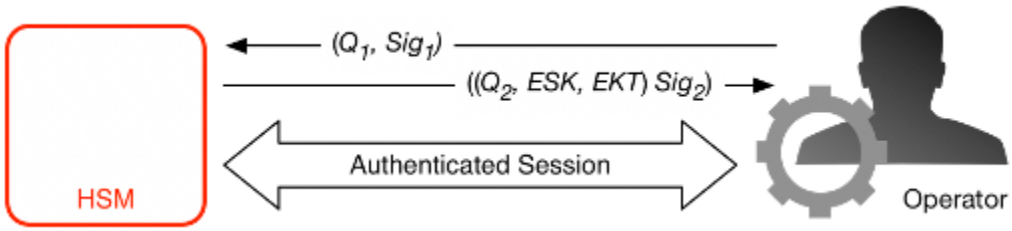
1. Per ogni firma, estrae la chiave pubblica del firmatario dallo stato del dominio e verifica la firma sul comando.
2. Verifica che il set di firmatari soddisfi una regola per il comando.

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 di autenticazione reciproca tra l'host del servizio ECDHE e l'host. HSM Lo scambio viene avviato dall'host del servizio e completato da. HSM Restituisce HSM inoltre 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 un membro del dominio e dispone di una coppia di chiavi per la firma dell'identità (d HOS_i, $QHOS_i$) e di una copia autentica delle chiavi pubbliche di identitàHSMs. Utilizza il suo set di chiavi per la firma dell'identità per negoziare in modo sicuro una chiave di sessione che può essere utilizzata tra l'host del servizio e qualsiasi parte del dominio. HSM Ai token chiave esportati è associato un periodo di validità, dopodiché è necessario negoziare una nuova chiave.



Il processo inizia con il riconoscimento da parte dell'host del servizio che richiede una chiave di sessione per inviare e ricevere flussi di comunicazione sensibili tra l'host e un HSM membro del dominio.

1. Un host di servizi genera una ECDH coppia di chiavi temporanea (d_1, Q_1) e la firma con la sua chiave di identità Sig1 = Sign (DoS, Q). $_1$
2. HSMVerifica la firma sulla chiave pubblica ricevuta utilizzando il token di dominio corrente e crea una coppia di ECDH chiavi temporanea ($_2$d, Q). $_2$ Quindi completa la procedura ECDH-key-exchange conforme alla Raccomandazione per gli schemi di definizione delle chiavi a coppie che utilizzano la crittografia a logaritmi discreti (rivista) per formare una chiave negoziata a 256 bit. AES GCM Genera HSM una nuova chiave di sessione a 256 bit. AES GCM Crittografa la

chiave di sessione con la chiave negoziata per formare la chiave di sessione crittografata (). ESK Inoltre, crittografa la chiave di sessione sotto la chiave di dominio come token chiave esportato. EKT Infine, firma un valore restituito con la sua coppia di chiavi di identità Sig_2 = Sign (dHSK, (Q_2,ESK,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 ECDH chiavi secondo la Raccomandazione per gli schemi di definizione delle chiavi a coppie che utilizzano la crittografia a logaritmi discreti (rivista). Successivamente decripta per ottenere la chiave di sessione SK. ESK

Durante il periodo di validità di EKT, l'host del servizio può utilizzare la chiave di sessione negoziata SK per inviare comandi crittografati tramite busta a. HSM Ogni service-host-initiated comando di questa sessione autenticata include. EKT HSMRisponde utilizzando la stessa chiave di sessione negoziata SK.

Processo di replica per chiavi multi-regione

AWS KMS utilizza un meccanismo di replica interregionale per copiare il materiale chiave in una KMS chiave da una chiave HSM in una Regione AWS a un'HSMaltra. Regione AWS Affinché questo meccanismo funzioni, la KMS chiave che viene replicata deve essere una chiave multiregionale. Quando si replica una KMS chiave da una regione all'altra, le HSMs regioni non possono comunicare direttamente, perché si trovano 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 una AWS KMS HSM viene firmato crittograficamente utilizzando una chiave di firma di replica. Le chiavi di firma della replica (RSKs) sono ECDSA chiavi sulla curva P-384. NIST Ogni regione ne possiede almeno una e RSK il componente pubblico di ciascuna RSK è condiviso con tutte le altre regioni della stessa partizione. AWS

Il processo di replica tra regioni per copiare il materiale chiave dalla regione A alla regione B funziona come segue:

1. La regione B HSM in genera una ECDH chiave temporanea sulla curva NIST P-384, Replication Agreement Key B (). RAKB Il componente pubblico di RAKB viene inviato a una regione HSM A dal servizio proxy.
2. Il componente HSM in Region A riceve il componente pubblico della curva P-384 RAKB e quindi genera un'altra ECDH chiave temporanea sulla curva NIST P-384, Replication Agreement Key A

(). RAKA HSMEsegue lo schema di definizione delle ECDH chiavi RAKA e il componente pubblico diRAKB, e ricava una chiave simmetrica dall'output, Replication Wrapping Key (). RWK RWKViene utilizzato per crittografare il materiale chiave della chiave KMS multiregionale che viene replicata.

3. Il componente pubblico RAKA e il materiale chiave crittografato con il RWK vengono inviati alla HSM regione B tramite il servizio proxy.
4. La HSM regione B riceve la componente pubblica RAKA e il materiale chiave crittografato utilizzando ilRWK. HSMDeriva RWK eseguendo lo schema di stabilimento ECDH chiave RAKB e la componente pubblica diRAKA.
5. HSMNella Regione B utilizzano il RWK per decrittografare il materiale chiave dalla Regione A.

Protezione della durabilità

L'ulteriore durabilità del servizio per le chiavi generate dal servizio è garantita dall'uso dell'archiviazione offline e non volatile multipla dei token HSMs di dominio esportati e dall'archiviazione ridondante delle chiavi crittografate. KMS Gli offline HSMs sono membri dei domini esistenti. 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 esistentiHSM.

Il design di durabilità ha lo scopo di proteggere tutte le KMS chiavi in una regione AWS in caso di perdita su larga scala delle chiavi online HSMs o del set di KMS chiavi archiviate nel nostro sistema di archiviazione principale. AWS KMS keys con chiavi importate, il materiale non è incluso nelle protezioni di resistenza offerte dalle altre chiavi. KMS In caso di guasto 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

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 notifiche sulle modifiche significative, iscriviti al RSS feed.

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

Modifica	Descrizione	Data
Aggiornamento funzionalità	È stato aggiunto il supporto per KMS le chiavi multiregionali nelle regioni della Cina.	21 novembre 2024
AWS aggiornamento gestito della politica	È stato aggiornato il ruolo AWSKeyManagementServiceMultiRegionKeysServiceRolePolicycollegato 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 DeleteImportedKeyM	23 luglio 2024

	aterialrichieste ImportKey Materiale.	
Modifica della quota	È stata aumentata la frequenza di richieste di operazioni crittografiche predefinite per KMS chiavi, RSA KMS chiavi e chiavi di crittografia simmetrica. ECC SM2 KMS	8 luglio 2024
Nuova caratteristica	Aggiunto un nuovo `KeyUsage` tipo `KEY_AGREEMENT` per le KMS chiavi NIST -recommended elliptic curve (ECC) e (solo per le regioni SM2 della Cina) 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 politica gestita	Sono state aggiunte nuove autorizzazioni `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` che consentono di AWS KMS monitorare le modifiche nel cluster VPC che contiene il AWS CloudHSM cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di errori.	10 novembre 2023
Aggiornamento funzionalità	È stato aggiunto il supporto per il parametro. `DryRun` API	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 utilizzar e 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 AWS le 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 KMS chiavi in ogni account e regione.	8 luglio 2022
Aggiornamento funzionalità	È stato aggiunto il supporto per HMAC KMS le chiavi in altro modo 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 chiavi e API operazioni che generano e verificano HMAC codici.	19 aprile 2022
Modifica della documentazione	Sostituisci il termine customer master key (CMK) con AWS KMS keyand KMSkey.	30 agosto 2021
Nuova caratteristica	È stato aggiunto il supporto per le chiavi multiregionali, un set di KMS chiavi interoperabili in diverse regioni che hanno lo stesso ID chiave e lo stesso materiale chiave. È possibile utilizzare le chiavi multi-reg ione per crittografare i dati in una Regione e decrittografare i dati in una Regione diversa.	8 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	È stato aggiunto il supporto per le policy VPC degli endpoint.	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 asimmetri che.	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	Spiega come utilizzare algoritmi ibridi di scambio di chiavi post-quantistici TLS per le chiamate a. AWS KMS	4 novembre 2019
Modifica della quota	Sono state aumentate le quote di risorse per alcune APIs aziende che gestiscono le chiavi. KMS	18 settembre 2019
Modifica della quota	Sono state modificate le quote di risorse per KMS chiavi, alias e concessioni per 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 IAM console. La console originari aoriginalecone le relative istruzioni persul'usocome utilizzarla saranno disponibi li 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 AWS Secrets Manager utilizza AWS KMS le chiavi per crittografare il valore segreto in un segreto.	13 luglio 2018
Nuovo contenuto	Spiega in che modo DynamoDB AWS KMS AWS KMS keys utilizza per supportare la sua opzione di crittografia lato server.	23 maggio 2018
Nuova caratteristica	Spiega come utilizzare un endpoint privato VPC a cui connettersi 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 KMS le chiavi 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 per le autorizzazioni e Chiavi di condizione.	5 luglio 2016
Update	Sono state aggiornate porzioni della documentazione nel capitolo KMSaccesso con chiavi e autorizzazioni.	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	11 aprile 2016

Modifica	Descrizione	Data
	concessione per migliorare la chiarezza e la precisione.	
Nuovo contenuto	È stata aggiunta la documentazione relativa a Consentire a più IAM presidi 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 politica chiave per migliorarne la chiarezza e la precisione.	17 febbraio 2016
Aggiornamento	Sono state aggiornate le pagine degli argomenti sulla gestione delle KMS chiavi per una maggiore chiarezza.	5 gennaio 2016
Nuovo contenuto	È stata aggiunta la documentazione relativa a Come si AWS CloudTrail usa AWS KMS.	18 novembre 2015
Nuovo contenuto	Sono state aggiunte istruzioni per Modificare una politica chiave.	18 novembre 2015
Aggiornamento	È stata aggiornata la documentazione sull'utilizzo di Amazon Relational Database Service AWS KMS.	18 novembre 2015
Nuovo contenuto	È stata aggiunta documentazione su Amazon WorkSpaces.	6 novembre 2015

Modifica	Descrizione	Data
Update	È stata aggiornata la pagina Politiche chiave in AWS KMS per migliorarne la chiarezza.	22 ottobre 2015
Nuovo contenuto	È stata aggiunta la documentazione su Eliminare un AWS KMS keys, inclusa la documentazione di supporto relativa a Creazione di un allarme e Determinare l'utilizzo passato di una KMS chiave.	15 ottobre 2015
Nuovo contenuto	È stata aggiunta la documentazione relativa a Determinare l'accesso a AWS KMS keys.	15 ottobre 2015
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

Modifica	Descrizione	Data
Nuovo contenuto	È stato aggiunto un nuovo codice Java di esempio che mostra l'utilizzo dell'oper azione `UpdateAlias` .	1° giugno 2015
Aggiornamento	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 documenta zione relativa a Come EMR utilizza Amazon AWS KMS.	28 gennaio 2015
Nuovo contenuto	È stata aggiunta documenta zione su Amazon WorkMail.	28 gennaio 2015
Nuovo contenuto	È stata aggiunta documenta zione sull'utilizzo di Amazon Relational Database Service AWS KMS.	6 gennaio 2015
Nuovo contenuto	È stata aggiunta documenta zione 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.