aws

Entwicklerhandbuch

AWS Key Management Service

AWS Key Management Service: Entwicklerhandbuch

Table of Contents

AWS Key Management Service

AWS Key Management Service (AWS KMS) ist ein verwalteter Service, der das Erstellen und Kontrollieren kryptografischer Schlüssel zum Schutz Ihrer Daten vereinfacht. AWS KMS verwendet Hardwaresicherheitsmodule (HSMs), um Ihre AWS KMS keys unter dem FIPS-140-2-Validierungsprogramm für kryptografische Module zu validieren und zu schützen. Die Regionen China (Peking) und China (Ningxia) unterstützen das FIPS-140-2-Validierungsprogramm für kryptografische Module nicht. AWS KMS verwendet OSCCA-zertifizierte HSMs zum Schutz von KMS-Schlüsseln in den China-Regionen.

AWS KMS ist in die meisten anderen AWS-Services integriert, die Ihre Daten verschlüsseln. AWS KMS ist auch in AWS CloudTrail integriert, um die Verwendung Ihrer KMS-Schlüssel für Prüfungs-, regulatorische und Compliance-Anforderungen zu protokollieren.

Sie können die AWS KMS-API verwenden, um KMS-Schlüssel und spezielle Funktionen, wie benutzerdefinierte Schlüsselspeicher zu erstellen und zu verwalten. Die KMS-Schlüssel können Sie in kryptografischen Operationen verwenden. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz.

Sie können Ihre AWS KMS keys erstellen und verwalten:

- Erstellen, Bearbeiten und Anzeigen von symmetrischen und asymmetrischen KMS-Schlüsseln einschließlich HMAC-Schlüsseln.
- Kontrollieren Sie den Zugriff auf Ihre KMS-Schlüssel mithilfe von Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen. AWS KMS unterstützt die attributbasierte Zugriffskontrolle (ABAC). Sie können Richtlinien auch mithilfe von Bedingungsschlüssel abstimmen.
- Erstellen, Löschen, Auflisten und Aktualisieren von Aliassen, Anzeigenamen für Ihre KMS-Schlüssel. Sie können auch Aliasse verwenden, den Zugriff auf Ihre KMS-Schlüssel zu kontrollieren.
- Markieren Ihrer KMS-Schlüssel zur Identifizierung, Automatisierung und Kostenverfolgung. Sie können auch Markierungen verwenden, den Zugriff auf Ihre KMS-Schlüssel zu kontrollieren.
- Aktivieren und Deaktivieren von KMS-Schlüsseln
- Aktivieren und deaktivieren sie die automatischen Rotation der Verschlüsselungsinformationen in einem KMS-Schlüssel.
- Löschen von KMS-Schlüsseln, um den Schlüssel-Lebenszyklus abzuschließen.

Sie können Ihre KMS-Schlüssel in kryptografischen Operationenverwenden. Beispiele finden Sie unter Programmieren der AWS KMS-API.

- Verschlüsseln, Entschlüsseln und erneutes Verschlüsseln von Daten mit symmetrischen oder asymmetrischen KMS-Schlüsseln.
- Signieren und Verifizieren von Nachrichten mit asymmetrischen KMS-Schlüsseln.
- Generieren von exportierbaren symmetrischen Datenschlüsseln und asymmetrischen Datenschlüsselpaaren.
- Generieren und überprüfen von HMAC-Codes.
- Generieren von Zufallszahlen für kryptografische Anwendungen.

Sie können die erweiterten Funktionen von AWS KMS nutzen.

- Erstellen Sie multiregionale Schlüssel, die wie Kopien desselben KMS-Schlüssels in verschiedenen AWS-Regionen fungieren.
- Importieren Sie kryptografisches Material in einen KMS-Schlüssel.
- Erstellen Sie KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher, der durch Ihren AWS CloudHSM-Cluster unterstützt wird.
- Erstellen Sie KMS-Schlüssel in einem externen Schlüsselspeicher, der durch Ihre kryptografischen Schlüssel außerhalb von AWS unterstützt wird.
- Direkte Verbindung zu AWS KMS über einen privaten Endpunkt in Ihrer VPC.
- Verwenden von Hybrid-Post-Quantum-TLS, um zukunftsgerichtete Verschlüsselung während der Übertragung für die von Ihnen an AWS KMS gesendeten Daten bereitzustellen.

Durch die Nutzung von AWS KMS erhalten Sie mehr Kontrolle über den Zugriff auf von Ihnen verschlüsselte Daten. Sie können die Schlüsselverwaltungsfunktionen und kryptografischen Funktionen direkt in Ihren Anwendungen oder über die in AWS KMS integrierten AWS-Services verwenden. Wenn Sie Anwendungen für AWS schreiben oder AWS-Services nutzen, gibt Ihnen AWS KMS die Kontrolle darüber, wer Ihre AWS KMS keys verwenden und Zugriff auf Ihre verschlüsselten Daten erhalten kann.

AWS KMS ist in AWS CloudTrail integriert, einem Service, der Protokolldateien an Ihren angegebenen Amazon-S3-Bucket liefert. Mit können CloudTrail Sie überwachen und untersuchen, wie und wann Ihre KMS-Schlüssel verwendet wurden und wer sie verwendet hat.

AWS KMS in AWS-Regionen

Die AWS-Regionen, in denen AWS KMS unterstützt wird, werden unter AWS Key Management Service-Endpunkte und Kontingente aufgeführt. Wenn ein AWS KMS-Feature in einer AWS-Region nicht unterstützt wird, die von AWS KMS unterstützt wird, wird in dem funktionsbezogenen Thema auf den regionalen Unterschied eingegangen.

AWS KMS – Preise

Wie bei anderen AWS-Produkten, erfordert die Verwendung von AWS KMS keine Verträge oder Mindestkäufe. Weitere Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service – Preise.

Service Level Agreement

AWS Key Management Service wird durch ein Service Level Agreement ergänzt, das unsere Richtlinie zur Serviceverfügbarkeit definiert.

Weitere Informationen

- Weitere Informationen zu den in AWS KMS verwendeten Begriffen und Konzepten finden Sie unter AWS KMS-Konzepte.
- Weitere Informationen zur AWS KMS-API finden Sie in der AWS Key Management Service-API-Referenz. Beispiele in verschiedenen Programmiersprachen finden Sie unter Programmieren der AWS KMS-API.
- Weitere Informationen zur Verwendung von AWS CloudFormation-Vorlagen zur Erstellung und Verwaltung von Schlüsseln und Aliasen finden Sie unter AWS KMS Ressourcen erstellen mit AWS CloudFormation und AWS Key Management ServiceRessourcentypenreferenz imAWS CloudFormation-Benutzerhandbuch.
- Detaillierte technische Informationen dazu, wie AWS KMS die Kryptografie verwendet und KMS-Schlüssel schützt, finden Sie unter AWS Key Management Service – Kryptographische Details. In der Dokumentation zu Kryptographischen Details wird nicht beschrieben, wie AWS KMS in den Regionen China (Peking) und China (Ningxia) funktioniert.
- Eine Liste mit AWS KMS-Endpunkten, einschließlich FIPS-Endpunkten, in jeder AWS-Region finden Sie unter Service-Endpunkte imAWS Key Management Service-Thema der Allgemeine AWS-Referenz.
- Hilfe zu Fragen über AWS KMS finden Sie im AWS Key Management Service Diskussionsforum.

AWS KMS in den AWS-SDKs

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

AWS KMS-Konzepte

Machen Sie sich mit den grundlegenden Begriffen und Konzepten in AWS Key Management Service (AWS KMS) vertraut und erfahren Sie, wie sie ineinandergreifen, um Ihre Daten besser zu schützen.

Themen

- AWS KMS keys
- Kundenschlüssel und AWS-Schlüssel
- KMS-Schlüssel zur symmetrischen Verschlüsselung
- Asymmetrische KMS-Schlüssel
- HMAC-KMS-Schlüssel
- Datenschlüssel
- Datenschlüsselpaare
- Aliasnamen
- Benutzerdefinierte Schlüsselspeicher
- Kryptografische Operationen
- Schlüsselkennungen (KeyId)
- Schlüsselmaterial
- Ursprung des Schlüsselmaterials
- Schlüsselspezifikation

- Schlüsselnutzung
- Envelope-Verschlüsselung
- Verschlüsselungskontext
- Schlüsselrichtlinie
- Gewährung
- Prüfung der KMS-Schlüsselnutzung
- Schlüsselverwaltungsinfrastruktur

AWS KMS keys

AWS KMS keys (KMS-Schlüssel) sind die wichtigsten Ressourcen in AWS KMS. Sie können einen KMS-Schlüssel zum Verschlüsseln, Entschlüsseln und erneuten Verschlüsseln von Daten verwenden. Es kann auch Datenschlüssel generieren, die Sie außerhalb von AWS KMS nutzen können. Normalerweise werden Sie symmetrische KMS-Schlüssel zur Verschlüsselung verwenden, aber Sie können auch asymmetrische KMS-Schlüssel zur Verschlüsselung oder Signierung erstellen und verwenden sowie HMAC-KMS-Schlüssel zur Erzeugung und Überprüfung von HMAC-Tags erstellen und verwenden.

> **Note**
>
> AWS KMS ersetzt den Begriff Kundenhauptschlüssel (CMK) durch AWS KMS key und KMS-Schlüssel. Das Konzept hat sich nicht geändert. Um abwärtsinkompatible Änderungen zu vermeiden, werden von AWS KMS einige Varianten dieses Begriffs beibehalten.

Ein AWS KMS key ist eine logische Darstellung eines kryptografischen Schlüssels. Ein KMS-Schlüssel enthält Metadaten wie die Schlüssel-ID, die Schlüsselspezifikation, die Schlüsselverwendung, das Erstellungsdatum, die Beschreibung und den Schlüsselstatus. Vor allem aber enthält er einen Verweis auf das Schlüsselmaterial, das verwendet wird, wenn Sie kryptografische Vorgänge mit dem KMS-Schlüssel durchführen.

Sie können einen KMS-Schlüssel mit kryptografischem Schlüsselmaterial erstellen, das in FIPS-validierten Hardware-Sicherheitsmodulen von AWS KMS generiert wurde. Das Schlüsselmaterial für symmetrische KMS-Schlüssel und die privaten Schlüssel für asymmetrische KMS-Schlüssel verlassen AWS KMS nie unverschlüsselt. Um Ihre KMS-Schlüssel zu verwenden oder zu verwalten, müssen Sie AWS KMS verwenden. Weitere Informationen zum Erstellen und Verwalten von KMS-

Schlüsseln finden Sie unter Schlüssel verwalten. Weitere Informationen zur Verwendung von KMS-Schlüssel finden Sie in der AWS Key Management Service-API-Referenz.

Standardmäßig erstellt AWS KMS das Schlüsselmaterial für einen KMS-Schlüssel. Sie können dieses Schlüsselmaterial nicht extrahieren, exportieren, anzeigen oder verwalten. Die einzige Ausnahme ist der öffentliche Schlüssel eines asymmetrischen Schlüsselpaars, den Sie für die Verwendung außerhalb von AWSexportieren können. Außerdem können Sie dieses Schlüsselmaterial nicht löschen. Sie müssen den KMS-Schlüssel löschen. Sie können jedoch Ihr eigenes Schlüsselmaterial in einen KMS-Schlüssel importieren oder einen benutzerdefinierten Schlüsselspeicher verwenden, um KMS-Schlüssel zu erstellen, die Schlüsselmaterial in Ihrem AWS CloudHSM-Cluster oder Schlüsselmaterial in einem externen Schlüsselmanager verwenden, den Sie außerhalb von AWS besitzen und verwalten.

AWS KMS unterstützt außerdem multiregionale Schlüssel, mit denen Sie Daten in einer AWS-Region verschlüsseln und in einer anderen AWS-Region entschlüsseln können.

Weitere Informationen zum Erstellen und Verwalten von KMS-Schlüsseln finden Sie unter Erste Schritte. Weitere Informationen zur Verwendung von KMS-Schlüssel finden Sie in der AWS Key Management Service-API-Referenz.

Kundenschlüssel und AWS-Schlüssel

Die von Ihnen erstellten KMS-Schlüssel sind kundenverwaltete Schlüssel. AWS-Services, die KMS-Schlüssel zur Entschlüsselung Ihrer Service-Ressourcen verwenden, erstellen oft Schlüssel für Sie. KMS-Schlüssel, die von AWS-Services in Ihrem AWS-Konto erstellt werden, sind Von AWS verwaltete Schlüssel. KMS-Schlüssel, die von AWS-Services in einem Servicekonto erstellt werden, sind AWS-eigene Schlüssel.

Typ des KMS-Schlüssels	Kann KMS-Schlüsselmetadaten anzeigen	Kann KMS-Schlüssel verwalten	Wird nur für mein AWS-Konto verwendet	Automatisches Rotieren	Preise
Kundenverwalteter Schlüssel	Ja	Ja	Ja	Optional. Jedes Jahr (ungefähr 365 Tage)	Monatliche Gebühr (anteilig stündlich)

Typ des KMS-Schlüssels	Kann KMS-Schlüsselmetadaten anzeigen	Kann KMS-Schlüssel verwalten	Wird nur für mein AWS-Konto verwendet	Automatisches Rotieren	Preise
					Gebühr pro Nutzung
Von AWS verwalteter Schlüssel	Ja	Nein	Ja	Erforderlich Jedes Jahr (ungefähr 365 Tage)	Keine monatliche Gebühr Gebühr pro Nutzung (einige AWS-Services übernehmen diese Gebühr für Sie)
AWS-eigener Schlüssel	Nein	Nein	Nein	Variiert	Keine Gebühren

Bei den mit AWS KMS integrierten AWS-Services gibt es Unterschiede hinsichtlich ihrer Unterstützung für KMS-Schlüssel. Einige AWS-Services verschlüsseln Ihre Daten standardmäßig mit einem AWS-eigener Schlüssel oder Von AWS verwalteter Schlüssel. Einige AWS-Services unterstützen kundenverwaltete Schlüssel. Andere AWS-Services unterstützen alle Arten von KMS-Schlüsseln, damit Sie die Benutzerfreundlichkeit eines AWS-eigener Schlüssel, die Sichtbarkeit eines Von AWS verwalteter Schlüssel oder die Kontrolle eines kundenverwalteten Schlüssels nutzen können. Ausführliche Informationen zu den Verschlüsselungsoptionen, die ein AWS-Service anbietet, finden Sie im Benutzerhandbuch oder im Entwicklerhandbuch für den Service unter dem Thema Verschlüsselung im Ruhezustand.

Kundenverwaltete Schlüssel

Die von Ihnen erstellten KMS-Schlüssel sind kundenverwaltete Schlüssel. Kundenverwaltete Schlüssel sind KMS-Schlüssel in Ihrem AWS-Konto, die Sie erstellen, besitzen und verwalten. Sie

haben die volle Kontrolle über diese KMS-Schlüssel. Dies gilt auch für die Festlegung und Verwaltung ihrer Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen, ihre Aktivierung und Deaktivierung, die Drehung ihrer Verschlüsselungsinformationen, das Hinzufügen von Tags, das Erstellen von Aliassen, die sich auf die KMS-Schlüssel beziehen, und das Einplanen der KMS-Schlüssel zum Löschen.

Kundenverwaltete Schlüssel werden auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) der AWS Management Console für AWS KMS angezeigt. Um einen vom Kunden verwalteten Schlüssel endgültig zu identifizieren, verwenden Sie die -DescribeKeyOperation. Für kundenverwaltete Schlüssel ist der Wert des `KeyManager`-Felds der `DescribeKey`-Antwort `CUSTOMER`.

Sie können Ihren kundenverwalteten Schlüssel in kryptografischen Operationen verwenden und ihre Verwendung in AWS CloudTrail-Protokollen prüfen. Darüber hinaus bieten Ihnen viele mit AWS KMS integrierte AWS-Services die Möglichkeit, einen kundenverwalteten Schlüssel zum Schutz der Daten anzugeben, die für Sie gespeichert und verwaltet werden.

Für kundenverwaltete Schlüssel fällt eine monatliche Gebühr sowie eine Gebühr für die über das kostenlose Kontingent hinausgehende Nutzung an. Sie werden auf die AWS KMS-Kontingente für Ihr Konto angerechnet. Weitere Informationen finden Sie unter AWS Key Management Service – Preise und Kontingente.

Von AWS verwaltete Schlüssel

Von AWS verwaltete Schlüssel sind KMS-Schlüssel in Ihrem Konto, die in Ihrem Namen von einem AWS-Service, der in AWS KMS integriert ist, erstellt, verwaltet und verwendet werden.

Bei einigen AWS-Services können Sie einen Von AWS verwalteter Schlüssel oder einen kundenverwalteten Schlüssel auswählen, um Ihre Ressourcen in diesem Service zu schützen. Im Allgemeinen ist, sofern Sie nicht verpflichtet sind, den Verschlüsselungsschlüssel zu kontrollieren, der Ihre Ressourcen schützt, Von AWS verwalteter Schlüssel eine gute Wahl. Sie müssen den Schlüssel oder seine Schlüsselrichtlinie nicht erstellen oder verwalten, und für einen Von AWS verwalteter Schlüssel gibt es keine monatliche Gebühr.

Sie haben die Berechtigung, die Von AWS verwaltete Schlüssel in Ihrem Konto anzuzeigen, ihre Schlüsselrichtlinien abzurufen und ihre Verwendung in AWS CloudTrail-Protokollen zu prüfen. Sie können aber keine Eigenschaften von Von AWS verwaltete Schlüssel ändern, sie rotieren, ihre Schlüsselrichtlinien ändern oder ihre Löschung planen. Auch können Sie Von AWS verwaltete Schlüssel nicht direkt in kryptografischen Operationen verwenden. Der Service, der sie erstellt, verwendet sie in Ihrem Namen.

Von AWS verwaltete Schlüssel erscheinen auf der Von AWS verwaltete Schlüssel-Seite der AWS Management Console für AWS KMS. Die meisten Von AWS verwaltete Schlüssel können auch anhand ihrer Aliasnamen im Format `aws/service-name`, wie z. B. `aws/redshift`, identifiziert werden. Verwenden Sie die -DescribeKeyOperationVon AWS verwaltete Schlüssel, um einen endgültig zu identifizieren. Für Von AWS verwaltete Schlüssel ist der Wert des `KeyManager`-Felds der `DescribeKey`-Antwort `AWS`.

Alle Von AWS verwaltete Schlüssel werden automatisch alle drei Jahre rotiert. Dieser Rotationsplan kann nicht geändert werden.

> **Note**
>
> Im Mai 2022 hat AWS KMS den Rotationszeitplan für Von AWS verwaltete Schlüssel von alle drei Jahre (ungefähr 1 095 Tage) auf jedes Jahr (ungefähr 365 Tage) geändert.
> Neue Von AWS verwaltete Schlüssel werden automatisch ein Jahr nach ihrer Erstellung und etwa jedes Jahr danach rotiert.
> Vorhandene Von AWS verwaltete Schlüssel werden automatisch ein Jahr nach ihrer letzten Rotation und danach jedes Jahr rotiert.

Es gibt keine monatliche Gebühr für Von AWS verwaltete Schlüssel. Für die Nutzung, die über den kostenlosen Bereich hinausgeht, können Gebühren anfallen, aber einige AWS-Services übernehmen diese Kosten für Sie. Weitere Informationen finden Sie im Benutzerhandbuch oder Entwicklerhandbuch für den Service unter dem Thema Verschlüsselung im Ruhezustand. Für Einzelheiten vgl. AWS Key Management Service-Preise.

Von AWS verwaltete Schlüssel werden nicht auf Ressourcenkontingente für die Anzahl der KMS-Schlüssel in jeder Region Ihres Kontos angerechnet. Wenn die KMS-Schlüssel jedoch im Namen eines Prinzipals in Ihrem Konto verwendet werden, werden sie auf die Anforderungskontingente angerechnet. Details hierzu finden Sie unter Kontingente.

AWS-eigene Schlüssel

AWS-eigene Schlüssel sind eine Sammlung von KMS-Schlüssel, die ein AWS-Service besitzt und für die Verwendung in mehreren AWS-Konten verwaltet. Obwohl AWS-eigene Schlüssel nicht in Ihrem AWS-Konto enthalten sind, kann ein AWS-Service einen AWS-eigener Schlüssel verwenden, um die Ressourcen in Ihrem Konto zu schützen.

Bei einigen AWS-Services können Sie einen AWS-eigener Schlüssel oder einen kundenverwalteten Schlüssel auswählen. Im Allgemeinen ist, sofern Sie den Verschlüsselungsschlüssel, der Ihre Ressourcen schützt, überwachen oder kontrollieren müssen, ein AWS-eigener Schlüssel eine gute Wahl. AWS-eigene Schlüssel sind vollkommen kostenlos (keine monatlichen Gebühren oder Nutzungsgebühren), sie werden nicht auf die AWS KMS-Kontingente Ihres Kontos angerechnet und sie sind benutzerfreundlich. Sie müssen den Schlüssel oder seine Schlüsselrichtlinie nicht erstellen oder pflegen.

Die Rotation der AWS-eigene Schlüssel unterscheidet sich je nach Service. Informationen zur Rotation eines bestimmten AWS-eigener Schlüssel finden Sie im Benutzerhandbuch oder Entwicklerhandbuch für den Service unter dem Thema Verschlüsselung im Ruhezustand.

KMS-Schlüssel zur symmetrischen Verschlüsselung

Wenn Sie einen AWS KMS key erstellen, erhalten Sie standardmäßig einen KMS-Schlüssel mit symmetrischer Verschlüsselung. Dies ist der grundlegende und am häufigsten verwendete KMS-Schlüsseltyp.

In AWS KMS stellt ein KMS-Schlüssel mit symmetrischer Verschlüsselung einen 256-Bit-AES-GCM-Verschlüsselungsschlüssel dar, außer in den China-Regionen, in denen er einen 128-Bit-SM4-Verschlüsselungsschlüssel darstellt. Symmetrisches Schlüsselmaterial lässt zu keiner Zeit AWS KMS unverschlüsselt. Zur Verwendung eines KMS-Schlüssels mit symmetrischer Verschlüsselung müssen Sie AWS KMS aufrufen. Symmetrische Verschlüsselungsschlüssel werden bei der symmetrischen Verschlüsselung verwendet, wobei der gleiche Schlüssel für die Verschlüsselung und Entschlüsselung verwendet wird. Sofern Ihre Aufgabe nicht explizit asymmetrische Verschlüsselung erfordert, sind KMS-Schlüssel mit symmetrischer Verschlüsselung, die AWS KMS niemals unverschlüsselt lassen, eine gute Wahl.

AWS-Services, die mit AWS KMS integriert sind, verwenden zum Verschlüsseln Ihrer Daten nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Diese Services unterstützen keine Verschlüsselung mit asymmetrischen KMS-Schlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Technisch gesehen ist die Schlüsselspezifikation für einen symmetrischen Schlüssel SYMMETRIC_DEFAULT, die Schlüsselverwendung ENCRYPT_DECRYPT und der Verschlüsselungsalgorithmus SYMMETRIC_DEFAULT. Details hierzu finden Sie unter Schlüsselspezifikation SYMMMETRIC_DEFAULT.

Sie können einen KMS-Schlüssel mit symmetrischer Verschlüsselung in AWS KMS verwenden, um Daten zu verschlüsseln, zu entschlüsseln und neu zu verschlüsseln sowie Datenschlüssel und Datenschlüsselpaare zu generieren. Sie können multiregionale KMS-Schlüssel mit symmetrischer Verschlüsselung erstellen, ihr eigenes Schlüsselmaterial in einen KMS-Schlüssel mit symmetrischer Verschlüsselung importieren und KMS-Schlüssel mit symmetrischer Verschlüsselung in benutzerdefinierten Schlüsselspeichern erstellen. Eine Tabelle mit den Operationen, die Sie auf unterschiedlichen KMS-Schlüsseltypen durchführen können, finden Sie unter Schlüsseltypreferenz.

Asymmetrische KMS-Schlüssel

Sie können asymmetrische KMS-Schlüssel in AWS KMS erstellen. Ein asymmetrischer KMS-Schlüssel repräsentiert einen mathematisch verwandtes Schlüsselpaar mit einem öffentlichen und einem privaten Schlüssel. Der private Schlüssel verlässt AWS KMS niemals unverschlüsselt. Um den privaten Schlüssel zu verwenden, müssen Sie AWS KMS aufrufen. Sie können den öffentlichen Schlüssel innerhalb von AWS KMS verwenden, indem Sie die AWS KMS-API-Operationen aufrufen, oder den öffentlichen Schlüssel herunterladen und ihn außerhalb von AWS KMS verwenden. Sie können auch multiregionale asymmetrische KMS-Schlüssel erstellen.

Sie können asymmetrische KMS-Schlüssel erstellen, die RSA-Schlüsselpaare oder SM2-Schlüsselpaare (nur China-Regionen) für die Verschlüsselung mit öffentlichem Schlüssel oder für die Signatur und Verifizierung darstellen, oder Elliptic-Curve-Schlüsselpaare für die Signatur und Verifizierung.

Weitere Informationen zum Erstellen und Verwenden von asymmetrischen KMS-Schlüsseln finden Sie unter Asymmetrische Schlüssel in AWS KMS.

HMAC-KMS-Schlüssel

Ein HMAC-KMS-Schlüssel stellt einen symmetrischen Schlüssel unterschiedlicher Länge dar, der zum Generieren und Überprüfen von Hash-basierten Nachrichtenauthentifizierungscodes (HMAC) verwendet wird. Dieses Schlüsselmaterial lässt AWS KMS zu keiner Zeit unverschlüsselt. Um einen HMAC-Schlüssel zu verwenden, rufen Sie die `GenerateMac`- oder `VerifyMac`-API-Operationen auf.

Sie können auch multiregionale HMAC-KMS-Schlüssel erstellen.

Weitere Informationen zum Erstellen und Verwenden von HMAC-KMS-Schlüsseln finden Sie unter HMAC-Schlüssel in AWS KMS.

Datenschlüssel

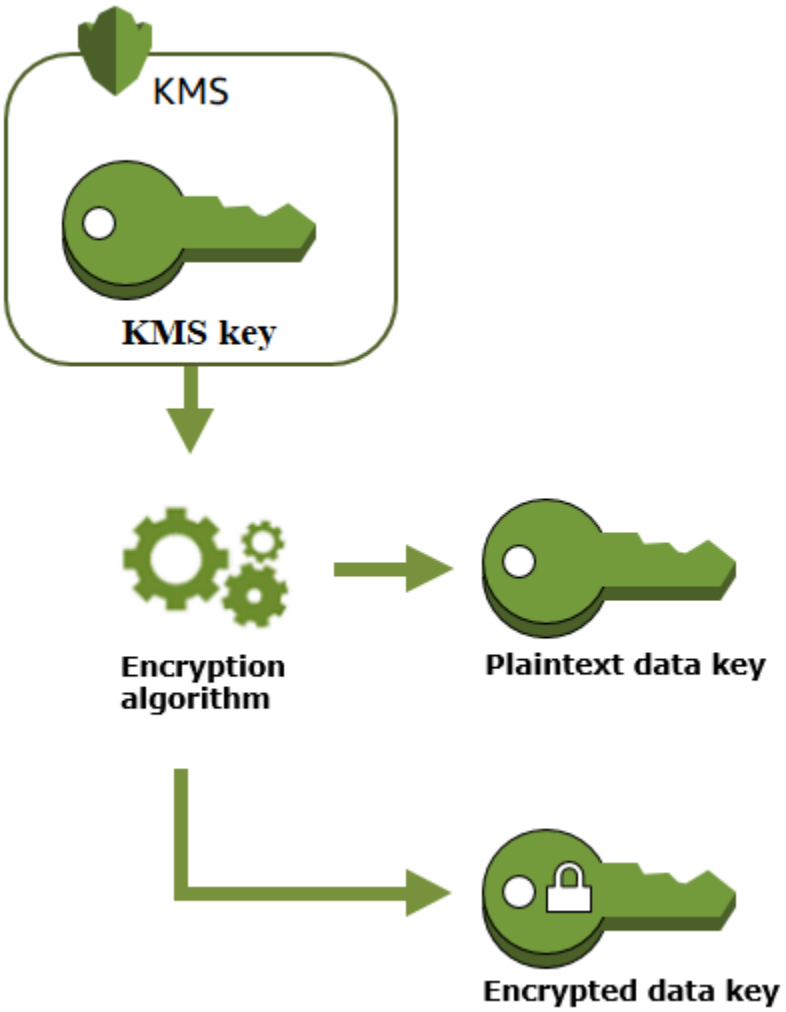
Datenschlüssel sind symmetrische Schlüssel, mit denen Sie Daten verschlüsseln können. Dazu gehören große Datenmengen und andere Datenverschlüsselungsschlüssel. Im Gegensatz zu symmetrischen KMS-Schlüsseln, die nicht heruntergeladen werden können, werden Datenschlüssel für die Verwendung außerhalb von AWS KMS zurückgegeben.

Wenn AWS KMS Datenschlüssel generiert, gibt es einen Klartextdatenschlüssel für die sofortige Verwendung zurück (optional) und eine verschlüsselte Kopie des Datenschlüssels, den Sie sicher mit den Daten speichern können. Wenn Sie bereit sind, die Daten zu entschlüsseln, fragen Sie zuerst AWS KMS, den verschlüsselten Datenschlüssel zu entschlüsseln.

AWS KMS generiert, verschlüsselt und entschlüsselt Datenschlüssel. AWS KMS speichert, verwaltet und verfolgt aber keine Datenschlüssel und führt auch keine kryptografischen Operationen mit Datenschlüsseln durch. Sie müssen Datenschlüssel außerhalb von AWS KMS verwenden und verwalten. Hilfe bei der sicheren Verwendung von Datenschlüsseln finden Sie unter AWS Encryption SDK.

Erstellen eines Datenschlüssels

Um einen Datenschlüssel zu erstellen, rufen Sie die -GenerateDataKeyOperation auf. AWS KMS generiert den Datenschlüssel. Anschließend wird eine Kopie des Datenschlüssels unter dem von Ihnen angegebenen KMS-Schlüssel mit symmetrischer Verschlüsselung verschlüsselt. Diese Operation gibt eine Klartextkopie des Datenschlüssels und die unter dem KMS-Schlüssel verschlüsselte Kopie des Datenschlüssels zurück. Die folgende Abbildung zeigt diese Operation.

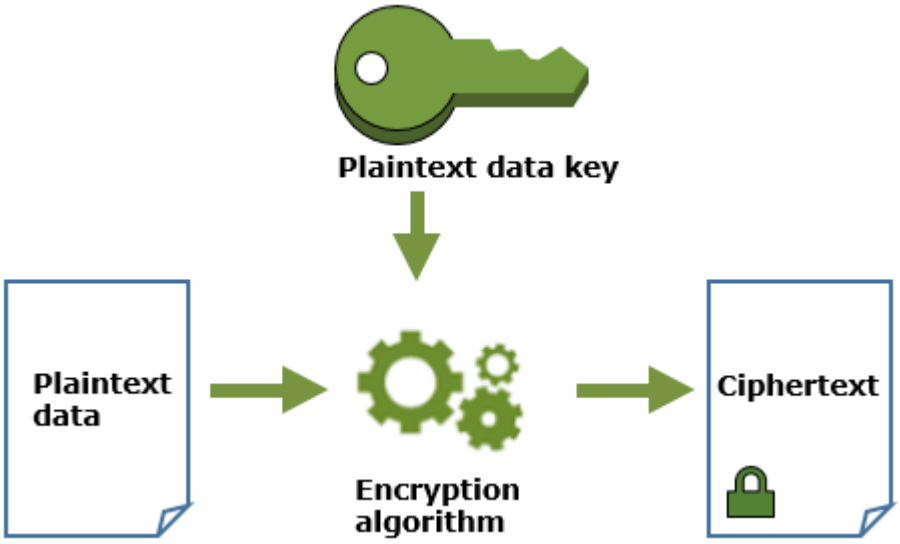


AWS KMS unterstützt auch die -GenerateDataKeyWithoutPlaintextOperation, die nur einen verschlüsselten Datenschlüssel zurückgibt. Wenn Sie den Datenschlüssel verwenden müssen, fordern Sie AWS KMS auf, ihn zu entschlüsseln.

Verschlüsseln von Daten mit einem Datenschlüssel

AWS KMS kann einen Datenschlüssel nicht zum Verschlüsseln von Daten verwenden. Sie können den Datenschlüssel jedoch außerhalb von AWS KMS verwenden, z. B. mit OpenSSL oder einer kryptografischen Bibliothek wie AWS Encryption SDK.

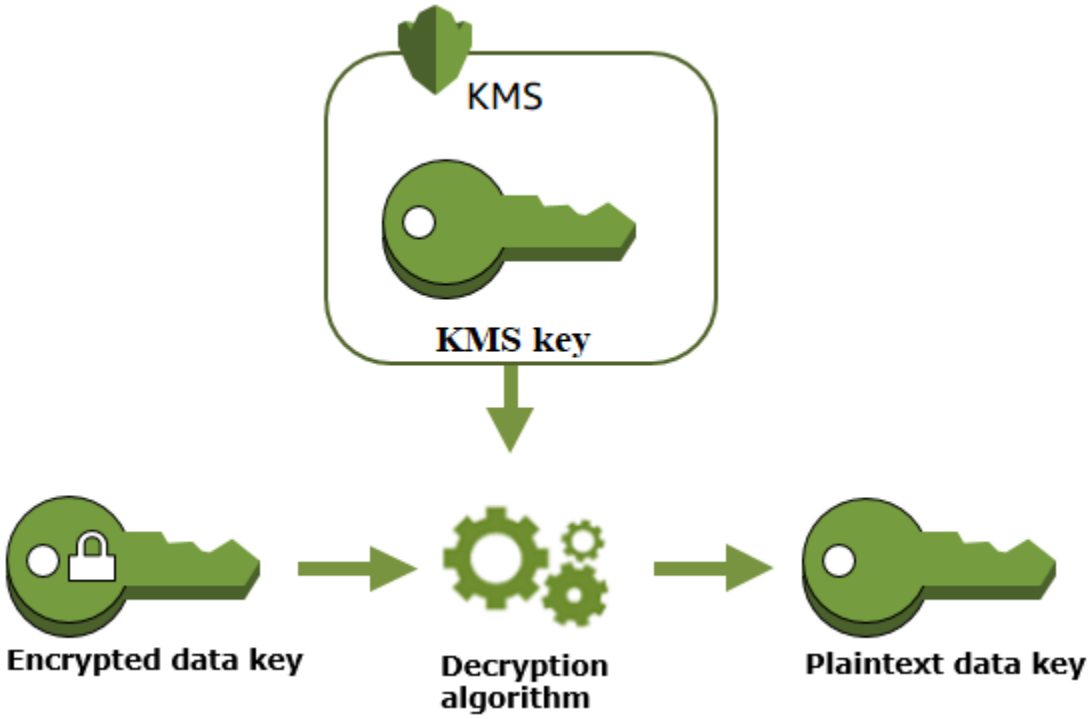
Entfernen Sie den Klartext-Datenschlüssel nach dem Verschlüsseln der Daten möglichst schnell aus dem Arbeitsspeicher. Sie können den verschlüsselten Datenschlüssel sicher zusammen mit den verschlüsselten Daten speichern, damit er zum Entschlüsseln der Daten verfügbar ist.



Entschlüsseln von Daten mit einem Datenschlüssel

Um Daten zu entschlüsseln, übergeben Sie den verschlüsselten Datenschlüssel an die Operation Decrypt. AWS KMS verwendet Ihren KMS-Schlüssel zum Entschlüsseln des Datenschlüssels und gibt dann den Klartext-Datenschlüssel zurück. Entschlüsseln Sie die Daten mit dem Klartext-Datenschlüssel und entfernen den Klartext-Datenschlüssel dann schnellstmöglich aus dem Arbeitsspeicher.

Das folgende Diagramm zeigt, wie Sie mit der Operation `Decrypt` einen verschlüsselten Datenschlüssel entschlüsseln.



Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel

Wenn ein KMS-Schlüssel unbrauchbar wird, wirkt sich das fast sofort aus (vorbehaltlich einer letztendlichen Konsistenz). Der Schlüsselstatus des KMS-Schlüssels ändert sich, um seinen neuen Zustand widerzuspiegeln, und alle Anforderungen der Verwendung des KMS-Schlüssels in kryptografischen Vorgängen schlagen fehl.

Die Auswirkungen auf die mit dem KMS-Schlüssel verschlüsselten Datenschlüssel und auf die mit dem Datenschlüssel verschlüsselten Daten werden jedoch so lange verzögert, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels.

KMS-Schlüssel können aus einer Vielzahl von Gründen unbrauchbar werden, einschließlich der folgenden Aktionen, die Sie möglicherweise durchführen:

- Deaktivieren des KMS-Schlüssels
- Planen der Löschung des KMS-Schlüssels
- Löschen des Schlüsselmaterials aus einem KMS-Schlüssel mit importiertem Schlüsselmaterial oder Ablaufenlassen des importierten Schlüsselmaterials
- Trennen des AWS CloudHSM-Schlüsselspeichers, der den KMS-Schlüssel hostet, oder Löschen des Schlüssels aus dem AWS CloudHSM-Cluster, der als Schlüsselmaterial für den KMS-Schlüssel dient

- Trennen des externen Schlüsselspeichers, der den KMS-Schlüssel hostet, oder jede andere Aktion, die Verschlüsselungs- und Entschlüsselungsanforderungen an den Proxy des externen Schlüsselspeichers beeinträchtigt, einschließlich der Löschung des externen Schlüssels aus seinem externen Schlüsselmanager

Dieser Effekt ist besonders wichtig für die vielen AWS-Services, die Datenschlüssel verwenden, um die vom Service verwalteten Ressourcen zu schützen. Das folgende Beispiel verwendet Amazon Elastic Block Store (Amazon EBS) und Amazon Elastic Compute Cloud (Amazon EC2). Verschiedene AWS-Services verwenden Datenschlüssel auf unterschiedliche Weise. Einzelheiten finden Sie im Abschnitt „Datenschutz“ des Kapitels „Sicherheit“ für den AWS-Service.

Betrachten Sie beispielsweise folgendes Szenario:

1. Sie erstellen ein verschlüsseltes EBS-Volume und geben einen KMS-Schlüssel an, um es zu schützen. Amazon EBS fordert AWS KMS auf, den KMS-Schlüssel zum Generieren eines verschlüsselten Datenschlüssels für das Volume zu verwenden. Amazon EBS speichert den verschlüsselten Datenschlüssel zusammen mit den Metadaten des Volumes.
2. Wenn Sie das EBS-Volume an eine EC2-Instance anfügen, verwendet Amazon EC2 Ihren KMS-Schlüssel zur Entschlüsselung des verschlüsselten Datenschlüssels des EBS-Volumes. Amazon EC2 verwendet den Datenschlüssel in der Nitro-Hardware, die für die Verschlüsselung aller Festplatten-I/Os auf dem EBS-Volume verantwortlich ist. Der Datenschlüssel bleibt in der Nitro-Hardware erhalten, während das EBS-Volume mit der EC2-Instance verbunden ist.
3. Sie führen eine Aktion aus, die den KMS-Schlüssel unbrauchbar macht. Dies hat keine unmittelbaren Auswirkungen auf die EC2-Instance oder das EBS-Volume. Amazon EC2 verwendet den Datenschlüssel, nicht den KMS-Schlüssel, um alle Festplatten-E/A zu verschlüsseln, während das Volume an die Instance angefügt ist.
4. Wenn jedoch das verschlüsselte EBS-Volume von der EC2-Instance getrennt wird, entfernt Amazon EBS den Datenschlüssel von der Nitro-Hardware. Wird das verschlüsselte EBS-Volume dann wieder an eine EC2-Instance angefügt, schlägt dies fehl, weil Amazon EBS nicht den KMS-Schlüssel verwenden kann, um den verschlüsselten Datenschlüssel des Volumes zu entschlüsseln. Um das EBS-Volume wieder zu verwenden, müssen Sie den KMS-Schlüssel wieder brauchbar machen.

Datenschlüsselpaare

Datenschlüsselpaare sind asymmetrische Datenschlüssel, die aus einem mathematisch verwandten öffentlichen Schlüssel und privaten Schlüssel bestehen. Sie sind zur Verwendung für clientseitige Verschlüsselung und Entschlüsselung oder Signatur und Verifizierung außerhalb von AWS KMS bestimmt.

Im Gegensatz zu den Datenschlüsselpaaren, die Tools wie OpenSSL generieren, schützt AWS KMS den privaten Schlüssel in jedem Datenschlüsselpaar unter einem KMS-Schlüssel mit symmetrischer Verschlüsselung in AWS KMS, den Sie angeben. AWS KMS speichert, verwaltet und verfolgt aber keine Datenschlüsselpaare und führt auch keine kryptografischen Operationen mit Datenschlüsselpaaren durch. Sie müssen Datenschlüsselpaare außerhalb von verwenden und verwalten AWS KMS.

AWS KMS unterstützt die folgenden Typen von Datenschlüsselpaaren:

- RSA-Schlüsselpaare: RSA_2048, RSA_3072 und RSA_4096
- Elliptic-Curve-Schlüsselpaare, ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521 und ECC_SECG_P256K1
- SM-Schlüsselpaare (nur China-Regionen): SM2

Welchen Typ von Datenschlüsselpaar Sie auswählen, hängt normalerweise von Ihrem Anwendungsfall oder den gesetzlichen Anforderungen ab. Die meisten Zertifikate erfordern RSA-Schlüssel. Als digitale Signaturen werden häufig Elliptic Curve (EC)-Schlüssel verwendet. ECC_SECG_P256K1-Schlüssel werden häufig für Kryptowährungen verwendet. AWS KMS empfiehlt, dass Sie ECC-Schlüsselpaare für die Signatur verwenden und RSA-Schlüsselpaare entweder für Verschlüsselung oder Signatur verwenden, aber nicht für beide. AWS KMS kann jedoch keine Beschränkungen für die Verwendung von Datenschlüsselpaaren außerhalb von AWS KMS erzwingen.

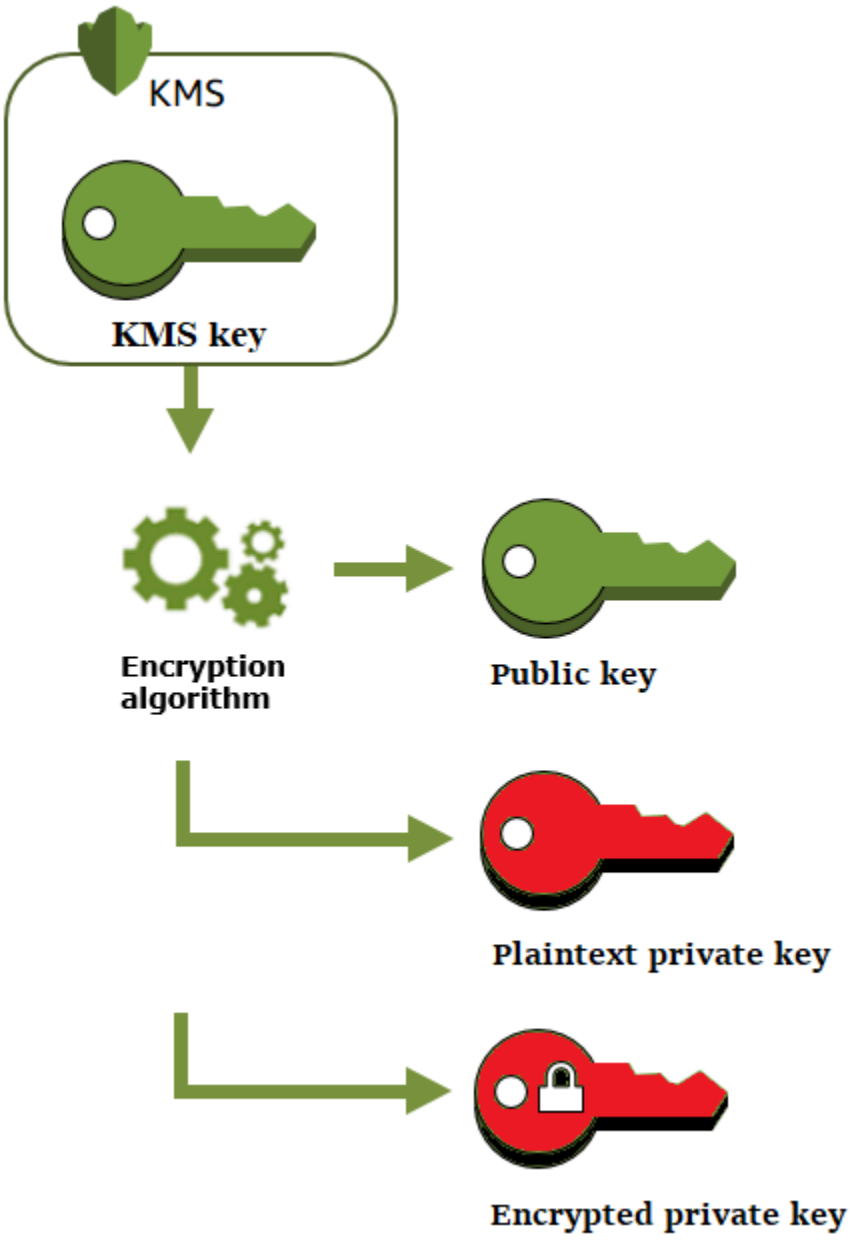
Erstellen eines Datenschlüsselpaars

Um ein Datenschlüsselpaar zu erstellen, rufen Sie die GenerateDataKeyPairWithoutPlaintext Operationen GenerateDataKeyPair oder auf. Geben Sie den KMS-Schlüssel mit symmetrischer Verschlüsselung an, den Sie zum Verschlüsseln des privaten Schlüssels verwenden möchten.

`GenerateDataKeyPair` gibt einen öffentlichen Klartextschlüssel, einen privaten Klartextschlüssel und einen verschlüsselten privaten Schlüssel zurück. Verwenden Sie diese Operation, wenn Sie umgehend einen privaten Klartextschlüssel benötigen, z. B. um eine digitale Signatur zu generieren.

`GenerateDataKeyPairWithoutPlaintext` gibt einen öffentlichen Klartextschlüssel und einen verschlüsselten privaten Schlüssel zurück, aber keinen privaten Klartextschlüssel. Verwenden Sie diese Operation, wenn Sie nicht sofort einen privaten Klartextschlüssel benötigen, z. B. wenn Sie mit einem öffentlichen Schlüssel verschlüsseln. Später, wenn Sie einen privaten Klartextschlüssel benötigen, um die Daten zu entschlüsseln, können Sie die Operation Decrypt aufrufen.

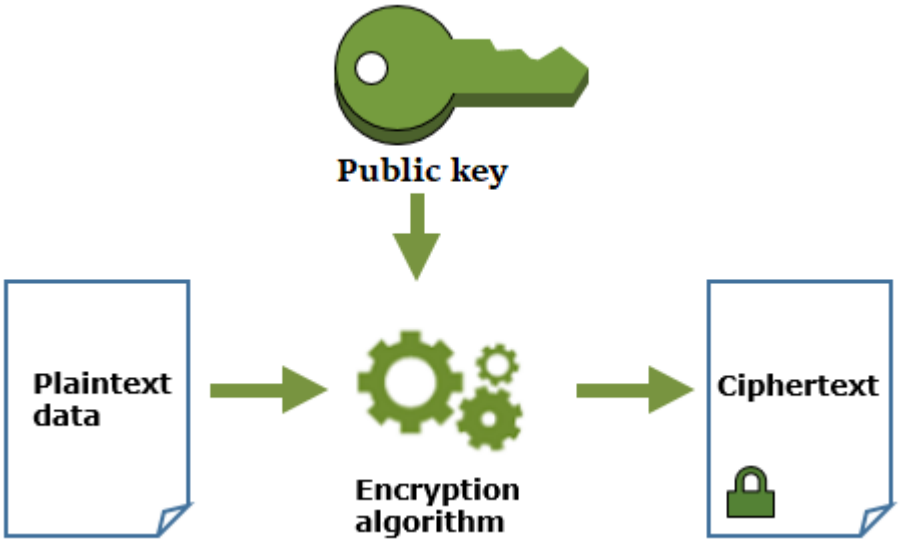
Die folgende Abbildung zeigt Operation `GenerateDataKeyPair`. Die Operation `GenerateDataKeyPairWithoutPlaintext` lässt den privaten Klartextschlüssel aus.



Verschlüsseln von Daten mit einem Datenschlüsselpaar

Wenn Sie mit einem Datenschlüsselpaar verschlüsseln, verwenden Sie den öffentlichen Schlüssel des Paares, um die Daten zu verschlüsseln, und den privaten Schlüssel desselben Paares, um die Daten zu entschlüsseln. In der Regel werden Datenschlüsselpaare verwendet, wenn viele Parteien Daten verschlüsseln müssen, die nur die Partei im Besitz des privaten Schlüssels entschlüsseln darf.

Die Parteien mit dem öffentlichen Schlüssel verwenden diesen Schlüssel, um Daten zu verschlüsseln, wie im folgenden Diagramm dargestellt.

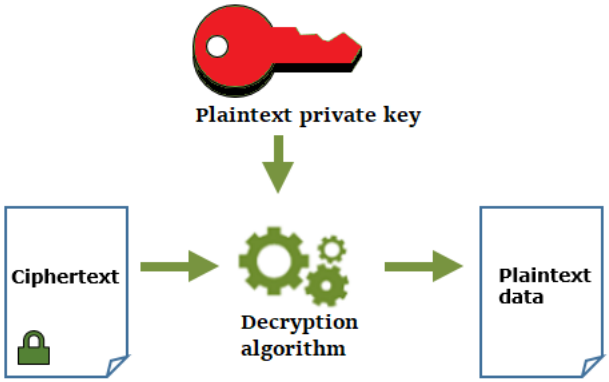


Entschlüsseln von Daten mit einem Datenschlüsselpaar

Um Ihre Daten zu entschlüsseln, verwenden Sie den privaten Schlüssel im Datenschlüsselpaar. Damit der Vorgang erfolgreich ausgeführt wird, müssen die öffentlichen und privaten Schlüssel aus demselben Datenschlüsselpaar stammen, und Sie müssen denselben Verschlüsselungsalgorithmus verwenden.

Um den verschlüsselten privaten Schlüssel zu entschlüsseln, übergeben Sie ihn an den Entschlüsselungsvorgang . Verwenden Sie den privaten Schlüssel, um die Daten zu entschlüsseln. Entfernen Sie dann den privaten Klartextschlüssel so schnell wie möglich aus dem Speicher.

Das folgende Diagramm zeigt, wie Sie den privaten Schlüssel in einem Datenschlüsselpaar verwenden, um Chiffretext zu entschlüsseln.



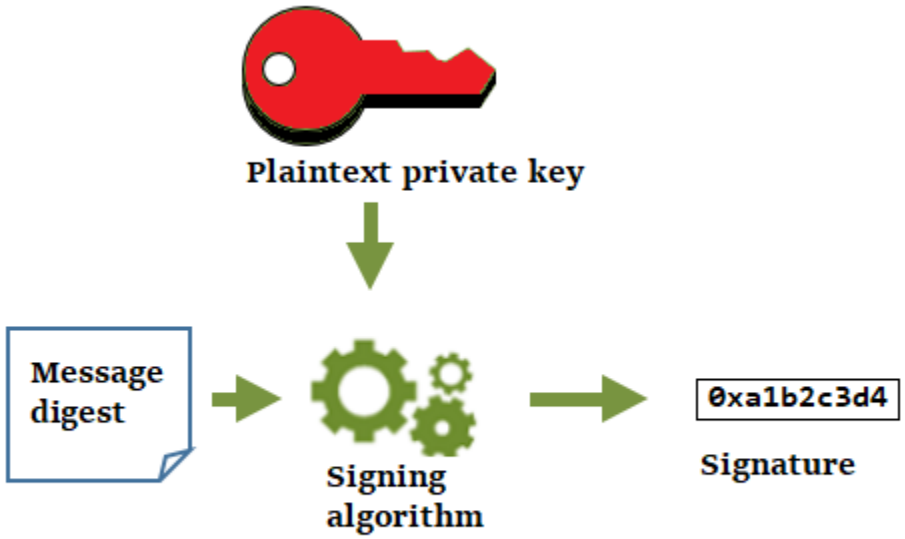
Signieren von Nachrichten mit einem Datenschlüsselpaar

Um eine kryptografische Signatur für eine Nachricht zu generieren, verwenden Sie den privaten Schlüssel im Datenschlüsselpaar. Jeder im Besitz des öffentlichen Schlüssels kann mit ihm überprüfen, dass die Nachricht mit Ihrem privaten Schlüssel signiert wurde und dass sie sich seit der Signatur nicht geändert hat.

Wenn Ihr privater Schlüssel verschlüsselt ist, übergeben Sie den verschlüsselten privaten Schlüssel an die Operation Decrypt. AWS KMS verwendet Ihren KMS-Schlüssel, um den Datenschlüssel zu entschlüsseln und gibt dann den privaten Klartextschlüssel zurück. Verwenden Sie den privaten Klartextschlüssel, um die Signatur zu generieren. Entfernen Sie dann den privaten Klartextschlüssel so schnell wie möglich aus dem Speicher.

Um eine Nachricht zu signieren, erstellen Sie einen Message Digest mit einer kryptografischen Hash-Funktion, z. B. dem Befehl `dgst` in OpenSSL. Übergeben Sie dann Ihren privaten Klartextschlüssel an den Signaturalgorithmus. Das Ergebnis ist eine Signatur, die den Inhalt der Nachricht darstellt. (Möglicherweise können Sie kürzere Nachrichten signieren, ohne vorher einen Digest zu erstellen. Die maximale Nachrichtengröße variiert je nach verwendetem Signatur-Tool.)

Das folgende Diagramm zeigt, wie Sie den privaten Schlüssel in einem Datenschlüsselpaar verwenden, um eine Nachricht zu signieren.

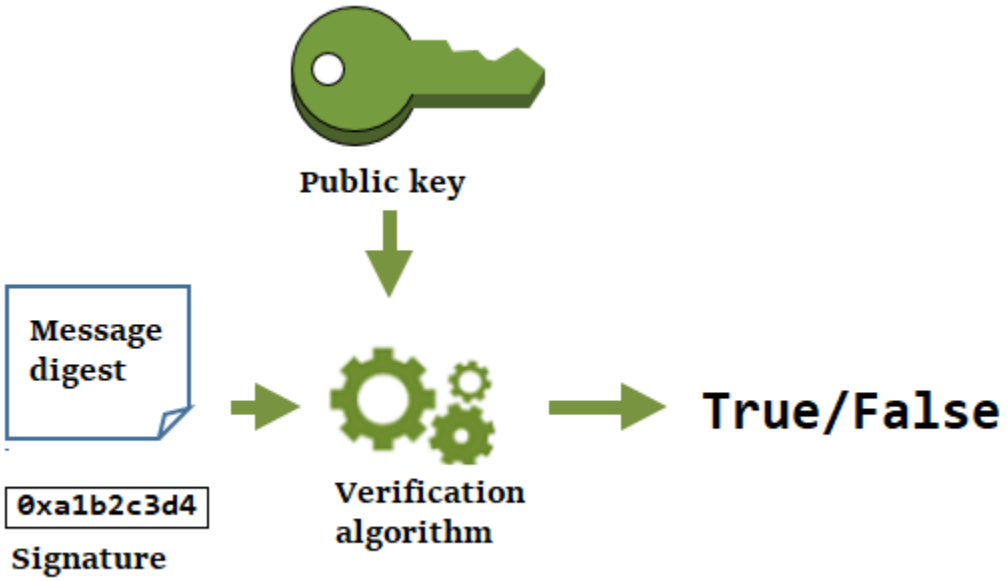


Überprüfen einer Signatur mit einem Datenschlüsselpaar

Jeder, der über den öffentlichen Schlüssel in Ihrem Datenschlüsselpaar verfügt, kann damit die Signatur überprüfen, die Sie mit Ihrem privaten Schlüssel generiert haben. Die Verifizierung bestätigt, dass ein autorisierter Benutzer die Nachricht mit dem angegebenen privaten Schlüssel und dem Signaturalgorithmus signiert hat und sich die Nachricht seit der Signatur nicht geändert hat.

Um erfolgreich zu sein, muss die Partei, die die Signatur überprüft, denselben Digest-Typ generieren, denselben Algorithmus verwenden und den öffentlichen Schlüssel verwenden, der dem privaten Schlüssel entspricht, der zum Signieren der Nachricht verwendet wird.

Das folgende Diagramm zeigt, wie der öffentliche Schlüssel in einem Datenschlüsselpaar verwendet wird, um eine Nachrichtensignatur zu überprüfen.



Aliasnamen

Verwenden eines Alias als Anzeigenamen für einen KMS-Schlüssel. Sie können beispielsweise auf einen KMS-Schlüssel als Test-Schlüsselverweisen, anstelle von 1234abcd-12ab-34cd-56ef-1234567890ab.

Aliasse erleichtern die Identifizierung eines KMS-Schlüssels in der AWS Management Console. Sie können einen Alias zum Identifizieren eines KMS-Schlüssels in einigen AWS KMS-Operationen verwenden, einschließlich kryptografische Operationen. In Anwendungen können Sie einen einzelnen Alias verwenden, um auf verschiedene KMS-Schlüssel in jedem AWS-Region zu verweisen.

Sie können den Zugriff auf KMS-Schlüssel auch auf Grundlage ihrer Aliasse zulassen und verweigern, ohne Richtlinien zu bearbeiten oder Berechtigungen zu verwalten. Diese Funktion ist Teil der AWS KMS-Unterstützung für Attribute-Based Access Control (ABAC, attributbasierte Zugriffssteuerung). Details hierzu finden Sie unter ABAC für AWS KMS.

In AWS KMS sind Aliasse unabhängige Ressourcen, keine Eigenschaften eines KMS-Schlüssels. Daher können Sie einen Alias hinzufügen, ändern und löschen, ohne den zugeordneten KMS-Schlüssel zu beeinflussen.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in einen Alias-Namen ein. Aliase können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Weitere Informationen:

- Ausführliche Informationen zu Aliassen finden Sie unter Verwenden von Aliassen.
- Informationen zu den Formaten von Schlüsselbezeichner, einschließlich Aliasse, finden Sie unter Schlüsselkennungen (KeyId).
- Hilfe zum Auffinden der Aliasse, die einem KMS-Schlüssel zugeordnet sind, finden Sie unter Suchen des Aliasnamens und des Alias-ARN
- Beispiele zur Erstellung und Verwaltung von Aliassen in verschiedenen Programmiersprachen finden Sie unter Arbeiten mit Aliasen.

Benutzerdefinierte Schlüsselspeicher

Ein benutzerdefinierter Schlüsselspeicher ist eine AWS KMS-Ressource, die von einem Schlüsselmanager außerhalb von AWS KMS unterstützt wird, den Sie besitzen und verwalten. Wenn Sie einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher für einen kryptografischen Vorgang verwenden, wird der kryptografische Vorgang tatsächlich in Ihrem Schlüsselmanager unter Verwendung seiner kryptografischen Schlüssel durchgeführt.

AWS KMS unterstützt AWS CloudHSM-Schlüsselspeicher, die durch einen AWS CloudHSM-Cluster unterstützt werden, und externe Schlüsselspeicher, die durch einen externen Schlüsselmanager außerhalb von AWS unterstützt werden.

Weitere Informationen finden Sie unter Benutzerdefinierte Schlüsselspeicher.

Kryptografische Operationen

In AWS KMS sind kryptografische Operationen API-Operationen, die KMS-Schlüssel zum Schutz von Daten verwenden. Da KMS-Schlüssel innerhalb von AWS KMS bleiben, müssen Sie AWS KMS aufrufen, um einen KMS-Schlüssel in einer kryptografischen Operation zu verwenden.

Um kryptografische Operationen mit KMS-Schlüssel durchzuführen, verwenden Sie die AWS-SDKs, AWS Command Line Interface (AWS CLI) oder die AWS Tools for PowerShell. Sie können keine kryptografischen Operationen in der AWS KMS-Konsole ausführen. Beispiele für das Aufrufen der kryptografischen Operationen in mehreren Programmiersprachen finden Sie unter Programmieren der AWS KMS-API.

In der folgenden Tabelle werden die kryptografischen Operationen von AWS KMS aufgeführt. Außerdem werden die Anforderungen an den Schlüsseltyp und die Schlüsselnutzung für KMS-Schlüssel angezeigt, die in der Operation verwendet werden.

Operation	Schlüsseltyp	Schlüsselnutzung
Decrypt	Symmetrisch oder asymmetrisch	`ENCRYPT_DECRYPT`
Encrypt	Symmetrisch oder asymmetrisch	`ENCRYPT_DECRYPT`
GenerateDataKey	Symmetrisch	`ENCRYPT_DECRYPT`
GenerateDataKeyPair	Symmetrisch [1] - nicht unterstützt bei KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern.	`ENCRYPT_DECRYPT`
GenerateDataKeyPairWithoutPlaintext	Symmetrisch [1] - nicht unterstützt bei KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern.	`ENCRYPT_DECRYPT`

Operation	Schlüsseltyp	Schlüsselnutzung
GenerateDataKeyWithoutPlaintext	Symmetrisch	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_ VERIFY_MAC
GenerateRandom	Bei dieser Operation wird kein KMS-Schlüssel verwendet.	N/A
ReEncrypt	Symmetrisch oder asymmetrisch	ENCRYPT_DECRYPT
Sign	Asymmetrisch	SIGN_VERIFY
Verify	Asymmetrisch	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_ VERIFY_MAC

[1] Generiert ein asymmetrisches Datenschlüsselpaar, das durch einen symmetrischen KMS-Schlüssel geschützt ist.

Informationen zu den Berechtigungen für kryptografische Operationen finden Sie unter the section called “Berechtigungsreferenz”.

Um AWS KMS reaktionsschnell und leistungsstark für alle Benutzer zu machen, legt AWS KMS Kontingente für die Anzahl der kryptografischen Operationen fest, die in jeder Sekunde aufgerufen werden können. Details hierzu finden Sie unter the section called “Gemeinsame Kontingente für kryptografische Operationen”.

Schlüsselkennungen (KeyId)

Schlüsselbezeichner fungieren als Namen für Ihre KMS-Schlüssel. Sie helfen Ihnen, Ihre KMS-Schlüssel in der Konsole zu erkennen. Sie verwenden sie, um anzugeben, welche KMS-Schlüssel Sie in AWS KMS-API-Operationen, Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen verwenden möchten. Die Schlüsselkennungswerte stehen in keinem Zusammenhang mit dem Schlüsselmaterial, das dem KMS-Schlüssel zugeordnet ist.

AWS KMS definiert mehrere Schlüsselbezeichner. Wenn Sie einen KMS-Schlüssel erstellen, generiert AWS KMS einen Schlüssel-ARN und eine Schlüssel-ID, die Eigenschaften des KMS-Schlüssels sind. Wenn Sie einen Alias erstellen, generiert AWS KMS einen Alias-ARN basierend auf dem von Ihnen definierten Aliasnamen. Sie können die Schlüssel- und Aliasbezeichner in der AWS Management Console und in der AWS KMS-API anzeigen.

In der AWS KMS-Konsole können Sie KMS-Schlüssel nach Schlüssel-ARN, Schlüssel-ID oder Aliasnamen anzeigen und filtern und nach Schlüssel-ID und Aliasnamen sortieren. Hilfestellung beim Suchen der Schlüsselbezeichner in der Konsole finden Sie unter the section called “Finden der Schlüssel-ID und des Schlüssel-ARN”.

In der AWS KMS-API werden die Parameter, die Sie zum Identifizieren eines KMS-Schlüssels verwenden, als `KeyId` oder eine Variante, z. B. `TargetKeyId` oder `DestinationKeyId`, bezeichnet. Die Werte dieser Parameter sind jedoch nicht auf Schlüssel-IDs beschränkt. Für einige eignet sich jeder gültige Schlüsselbezeichner. Informationen zu den Werten für jeden Parameter finden Sie in der Parameterbeschreibung in der API-Referenz für AWS Key Management Service.

> **Note**
>
> Achten Sie bei der Verwendung der AWS KMS-API auf die von Ihnen verwendete Schlüsselkennung. Unterschiedliche APIs erfordern unterschiedliche Schlüsselbezeichner. Verwenden Sie im Allgemeinen die vollständigste Schlüsselbezeichnung, die für Ihre Aufgabe praktisch ist.

AWS KMS unterstützt die folgenden Schlüsselbezeichner.

Schüssel-ARN

Der Schlüssel-ARN ist der Amazon-Ressourcenname (ARN) eines KMS-Schlüssels. Er ist eine eindeutige, vollqualifizierte Kennung für den KMS-Schlüssel. Ein Schlüssel-ARN enthält das AWS-Konto, die Region und die Schlüssel-ID. Hilfestellung beim Suchen des Schlüssel-ARN eines KMS-Schlüssels finden Sie unter the section called “Finden der Schlüssel-ID und des Schlüssel-ARN”.

Das Format eines Schlüssel-ARN lautet wie folgt:

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

Es folgt ein Beispiel eines Schlüssel-ARNs für einen einzelregionalen KMS-Schlüssel.

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

Das *Schlüssel-ID*-Element der Schlüssel-ARNs von multiregionalen Schlüsseln mit dem Präfix `mrk-`. Es folgt ein Beispiel eines Schlüssel-ARN für einen multiregionalen KMS-Schlüssel.

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

Schlüssel-ID

Die Schlüssel-ID identifiziert einen KMS-Schlüssel innerhalb eines Kontos und einer Region eindeutig. Hilfestellung beim Suchen der Schlüssel-ID eines KMS-Schlüssels finden Sie unter the section called “Finden der Schlüssel-ID und des Schlüssel-ARN”.

Es folgt ein Beispiel einer Schlüssel-ID für einen einzelregionalen KMS-Schlüssel.

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

Die Schlüssel-IDs von multiregionalen Schlüsseln beginnen mit dem Präfix `mrk-`. Es folgt ein Beispiel einer Schlüssel-ID für einen multiregionalen KMS-Schlüssel.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

Alias-ARN

Der Alias-ARN ist der Amazon-Ressourcenname (ARN) eines AWS KMS-Alias. Es ist ein eindeutiger, vollqualifizierter Bezeichner für den Alias und für den KMS-Schlüssel, den er repräsentiert. Ein Alias-ARN enthält das AWS-Konto, die Region und den Aliasnamen.

Ein Alias-ARN identifiziert zu einem bestimmten Zeitpunkt einen bestimmten KMS-Schlüssel. Da Sie jedoch den KMS-Schlüssel ändern können, der dem Alias zugeordnet ist, kann der Alias-ARN zu verschiedenen Zeiten unterschiedliche KMS-Schlüssel identifizieren. Hilfestellung beim Suchen des Alias-ARN eines KMS-Schlüssels finden Sie unter Suchen des Aliasnamens und des Alias-ARN.

Das Format eines Alias-ARN lautet wie folgt:

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

Es folgt der Alias-ARN für einen fiktiven `ExampleAlias`.

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

Alias-Name

Der Aliasname besteht aus einer Zeichenfolge mit bis zu 256 Zeichen. Er identifiziert einen zugeordneten KMS-Schlüssel innerhalb eines Kontos und einer Region eindeutig. In der AWS KMS-API beginnen Aliasnamen immer mit `alias/`. Hilfestellung beim Suchen des Aliasnamens eines KMS-Schlüssels finden Sie unter Suchen des Aliasnamens und des Alias-ARN.

Das Format eines Aliasnamens lautet wie folgt:

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

Zum Beispiel:

```
alias/ExampleAlias
```

Das `aws/`-Präfix für einen Aliasnamen ist für Von AWS verwaltete Schlüssel reserviert. Sie können keinen Alias mit diesem Präfix erstellen. Der Aliasname des Von AWS verwalteter Schlüssel für Amazon Simple Storage Service (Amazon S3) ist der Folgende:

```
alias/aws/s3
```

Schlüsselmaterial

Schlüsselmaterial ist die Zeichenfolge von Bits, die in einem kryptografischen Algorithmus verwendet wird. Geheimes Schlüsselmaterial muss geheim gehalten werden, um die kryptografischen Operationen, die es verwenden, zu schützen. Öffentliches Schlüsselmaterial ist für die gemeinsame Nutzung bestimmt.

Jeder KMS-Schlüssel enthält in seinen Metadaten einen Verweis auf sein Schlüsselmaterial. Die Herkunft des Schlüsselmaterials des KMS-Schlüssels mit symmetrischer Verschlüsselung kann variieren. Sie können Schlüsselmaterial verwenden, das AWS KMS generiert, Schlüsselmaterial, das

in dem AWS CloudHSM-Cluster eines benutzerdefinierten Schlüsselspeichers generiert wird, oder Ihr eigenes Schlüsselmaterial importieren. Wenn Sie AWS KMS-Schlüsselmaterial für Ihren KMS-Schlüssel mit symmetrischer Verschlüsselung verwenden, können Sie die automatische Drehung Ihres Schlüsselmaterials aktivieren.

Standardmäßig verfügt jeder KMS-Schlüssel über eindeutiges Schlüsselmaterial. Sie können jedoch eine Reihe von multiregionalen Schlüsseln mit demselben Schlüsselmaterial erstellen.

Ursprung des Schlüsselmaterials

Ursprung des Schlüsselmaterials ist eine KMS-Schlüssel-Eigenschaft, die die Quelle des Schlüsselmaterials im KMS-Schlüssel identifiziert. Sie wählen den Ursprung des Schlüsselmaterials, wenn Sie den KMS-Schlüssel erstellen. Danach kann er nicht mehr geändert werden. Die Quelle des Schlüsselmaterials wirkt sich auf die Sicherheit, Dauerhaftigkeit, Verfügbarkeit, Latenzzeit und Durchsatzmerkmale des KMS-Schlüssels aus.

Um den Ursprung des Schlüsselmaterials eines KMS-Schlüssels zu finden, verwenden Sie die -DescribeKeyOperation oder sehen Sie sich den Ursprungswert auf der Registerkarte Kryptografische Konfiguration der Detailseite für einen KMS-Schlüssel in der AWS KMS Konsole an. Hilfe finden Sie unter Anzeigen von Schlüsseln.

KMS-Schlüssel können einen der folgenden Werte für den Ursprung des Schlüsselmaterials haben.

`AWS_KMS`

AWS KMS erstellt und verwaltet das Schlüsselmaterial für den KMS-Schlüssel in einem eigenen Schlüsselspeicher. Dies ist der Standardwert und der empfohlene Wert für die meisten KMS-Schlüssel.

Hilfestellung beim Erstellen von Schlüsseln mit Schlüsselmaterial von AWS KMS finden Sie unter Erstellen von Schlüsseln.

`EXTERNAL (Import key material)`

Der KMS-Schlüssel enthält importiertes Schlüsselmaterial. Wenn Sie einen KMS-Schlüssel mit dem Ursprung des Schlüsselmaterials `External` erstellen, verfügt der KMS-Schlüssel über kein Schlüsselmaterial. Später können Sie Schlüsselmaterial in den KMS-Schlüssel importieren. Wenn Sie importiertes Schlüsselmaterial verwenden, müssen Sie dieses Schlüsselmaterial außerhalb von AWS KMS sichern und verwalten, einschließlich des Austauschs des Schlüsselmaterials, falls es abläuft. Details hierzu finden Sie unter Informationen zu importiertem Schlüsselmaterial.

Hilfestellung beim Erstellen eines KMS-Schlüssel für importiertes Schlüsselmaterial finden Sie unter Schritt 1: Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial.

`AWS_CLOUDHSM`

AWS KMS erstellt das Schlüsselmaterial im AWS CloudHSM-Cluster für Ihren AWS CloudHSM-Schlüsselspeicher.

Hilfe zum Erstellen eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher finden Sie unter Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher.

`EXTERNAL_KEY_STORE`

Das Schlüsselmaterial ist ein kryptografischer Schlüssel in einem externen Schlüsselmanager außerhalb von AWS. Dieser Ursprung wird nur für KMS-Schlüssel in einem externen Schlüsselspeicher unterstützt.

Hilfe zum Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher finden Sie unter Erstellen von KMS-Schlüsseln in einem externen Schlüsselspeicher.

Schlüsselspezifikation

Die Schlüsselspezifikation ist eine Eigenschaft, die die kryptografische Konfiguration des KMS-Schlüssels repräsentiert. Die Bedeutung der Schlüsselspezifikation unterscheidet sich vom Schlüsseltyp.

- AWS KMS-Schlüssel – Die Schlüsselspezifikation gibt an, ob der KMS-Schlüssel symmetrisch oder asymmetrisch ist. Sie bestimmt auch die Art des Schlüsselmaterials und die unterstützten Algorithmen. Sie wählen die Schlüsselspezifikation, wenn Sie den KMS-Schlüssel erstellen. Sie kann danach nicht mehr geändert werden. Die Standardschlüsselspezifikation, SYMMETRIC_DEFAULT, steht für einen symmetrischen 256-Bit-Verschlüsselungsschlüssel.

  > **Note**
  >
  > Der `KeySpec` für einen KMS-Schlüssel wird als `CustomerMasterKeySpec` bezeichnet. Der -`CustomerMasterKeySpec`Parameter der -CreateKeyOperation ist veraltet. Verwenden Sie stattdessen den `KeySpec`-Parameter, der auf dieselbe Weise funktioniert. Um Breaking Changes zu vermeiden, enthält die Antwort der DescribeKey Operationen `CreateKey` und jetzt sowohl - als auch `KeySpec` -`CustomerMasterKeySpec`Elemente mit denselben Werten.

Eine Liste der Schlüsselspezifikationen und Hilfe bei der Auswahl einer Schlüsselspezifikation finden Sie unter Auswählen der Schlüsselspezifikation. Um die Schlüsselspezifikation eines KMS-Schlüssels zu finden, verwenden Sie die -DescribeKeyOperation oder sehen Sie sich die Registerkarte Kryptografische Konfiguration auf der Detailseite für einen KMS-Schlüssel in der -AWS KMSKonsole an. Hilfe finden Sie unter Anzeigen von Schlüsseln.

Um die Schlüsselspezifikationen einzuschränken, die Prinzipale beim Erstellen von KMS-Schlüsseln verwenden können, verwenden Sie den Bedingungsschlüssel kms:KeySpec. Sie können mit dem Bedingungsschlüssel `kms:KeySpec` es Prinzipalen auch erlauben, AWS KMS-Operationen nur für KMS-Schlüssel mit einer bestimmen Schlüsselspezifikation aufzurufen. Beispielsweise können Sie die Berechtigung zum Planen des Löschens eines KMS-Schlüssels mit einer `RSA_4096`-Schlüsselspezifikation verweigern.

- Datenschlüssel (GenerateDataKey) – Die Schlüsselspezifikation bestimmt die Länge eines AES-Datenschlüssels.
- Datenschlüsselpaare (GenerateDataKeyPair) – Die Schlüsselpaarspezifikation bestimmt den Typ des Schlüsselmaterials im Datenschlüsselpaar.

Schlüsselnutzung

Die Schlüsselnutzung ist eine Eigenschaft, die bestimmt, welche kryptografischen Vorgänge der Schlüssel unterstützt. KMS-Schlüssel können die Schlüsselnutzung `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, oder `GENERATE_VERIFY_MAC` haben. Jeder KMS-Schlüssel kann nur eine Schlüsselnutzung haben. Die Verwendung eines KMS-Schlüssels für mehr als eine Art von Operation macht das Produkt beider Operationen anfälliger gegenüber Angriffen.

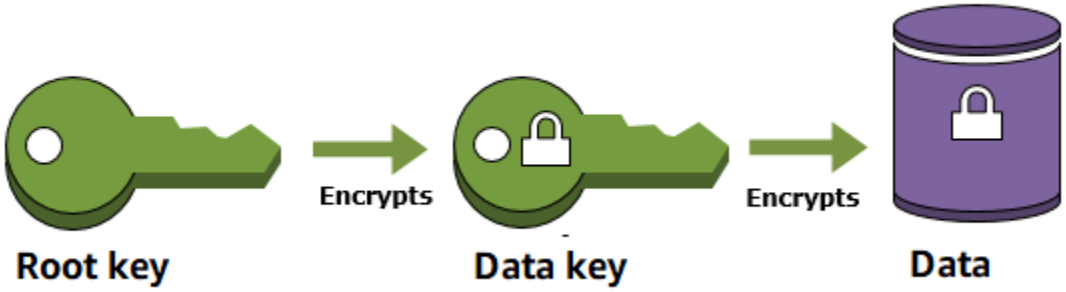
Hilfe bei der Auswahl der Schlüsselverwendung für Ihren KMS-Schlüssel finden Sie unter Auswählen der Schlüsselnutzung. Um die Schlüsselnutzung eines KMS-Schlüssels zu ermitteln, verwenden Sie die -DescribeKeyOperation oder wählen Sie die Registerkarte Kryptografische Konfiguration auf der Detailseite für einen KMS-Schlüssel in der AWS KMS Konsole. Hilfe finden Sie unter Anzeigen von Schlüsseln.

Envelope-Verschlüsselung

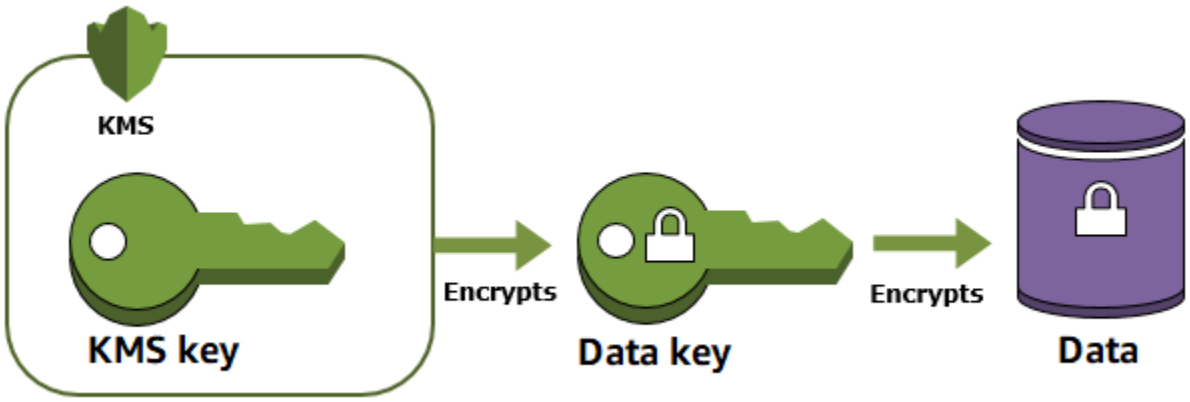
Wenn Sie Daten verschlüsseln, sind die Daten geschützt. Sie müssen aber jetzt den Verschlüsselungsschlüssel schützen. Eine Strategie besteht darin, ihn zu verschlüsseln. Envelope-

Verschlüsselung bezeichnet das Verschlüsseln von Klartextdaten mit einem Datenschlüssel und die anschließende Verschlüsselung des Datenschlüssels mit einem anderen Schlüssel.

Sie können auch den Datenverschlüsselungsschlüssel mit einem anderen Verschlüsselungsschlüssel verschlüsseln und diesen Verschlüsselungsschlüssel mit einem weiteren Verschlüsselungsschlüssel verschlüsseln. Doch schließlich muss ein Schlüssel als Klartext verbleiben, damit Sie die Schlüssel und die Daten entschlüsseln können. Dieser Top-Level-Klartext-Verschlüsselungsschlüssel zum Verschlüsseln von Schlüsseln wird als Stammschlüssel bezeichnet.



AWS KMS hilft Ihnen dabei, Verschlüsselungsschlüssel zu schützen, indem diese sicher gespeichert und verwaltet werden. Root-Schlüssel, die in AWS KMS gespeichert sind, bekannt als AWS KMS keys, verlassen niemals unverschlüsselt die FIPS-validierten Hardware-Sicherheitsmodule von AWS KMS. Um einen KMS-Schlüssel zu verwenden, müssen Sie AWS KMS aufrufen.



Envelope-Verschlüsselung bietet mehrere Vorteile:

- Schutz von Datenschlüsseln

  Wenn Sie einen Datenschlüssel verschlüsseln, müssen Sie sich nicht darum sorgen, wo Sie den verschlüsselten Schlüssel speichern, da die Sicherheit dieses Datenschlüssels inhärent durch

Verschlüsselung geschützt ist. Sie können den verschlüsselten Datenschlüssel sicher neben den verschlüsselten Daten speichern.

- Verschlüsselung der gleichen Daten unter mehreren Schlüsseln

  Verschlüsselungsoperationen können zeitaufwendig sein, insbesondere wenn die zu verschlüsselnden Daten große Objekte sind. Statt Rohdaten mit verschiedenen Schlüsseln mehrmals neu zu verschlüsseln, können Sie einfach die Datenschlüssel, die die Rohdaten schützen, neu verschlüsseln.

- Kombination der Stärken mehrerer Algorithmen

  Im Allgemeinen sind symmetrische Schlüsselalgorithmen schneller und erzeugen kleinere Verschlüsselungstexte als öffentliche Schlüsselalgorithmen. Algorithmen mit öffentlichem Schlüssel unterstützen jedoch eine inhärente Rollentrennung und eine einfachere Schlüsselverwaltung. Die Envelope-Verschlüsselung vereint die Stärken aller Strategien.

Verschlüsselungskontext

Alle kryptografischen Operationen von AWS KMS mit KMS-Schlüsseln mit symmetrischer Verschlüsselung akzeptieren einen Verschlüsselungskontext, einen optionalen Satz von nicht geheimen Schlüssel-Wert-Paaren, die zusätzliche Kontextinformationen zu den Daten enthalten können. AWS KMS verwendet den Verschlüsselungskontext als zusätzliche authentifizierte Daten (AAD) zur Unterstützung der authentifizierten Verschlüsselung.

Wenn Sie einen Verschlüsselungskontext in eine Verschlüsselungsanforderung einfügen, wird er kryptographisch an den Verschlüsselungstext gebunden. Daher ist zum Entschlüsseln (oder Entschlüsseln und erneuten Verschlüsseln) der Daten derselbe Verschlüsselungskontext erforderlich. Wenn der in der Entschlüsselungsanforderung angegebene Verschlüsselungskontext keine exakte Übereinstimmung ist (inkl. Groß-/Kleinschreibung), schlägt die Entschlüsselungsanforderung fehl. Nur die Reihenfolge der Schlüssel-Wert-Paare im Verschlüsselungskontext kann variieren.

> **Note**
>
> Sie können mit einem asymmetrische KMS-Schlüssel oder einem HMAC-KMS-Schlüssel keinen Verschlüsselungskontext in einer kryptografischen Produktion angeben. Asymmetrische Algorithmen und MAC-Algorithmen unterstützen keinen Verschlüsselungskontext.

Der Verschlüsselungskontext ist nicht geheim und nicht verschlüsselt. Er erscheint als Klartext in AWS CloudTrail-Protokollen, kann also zum Identifizieren und Kategorisieren der kryptografischen Operationen herangezogen werden. Ihr Verschlüsselungskontext sollte keine sensiblen Informationen enthalten. Wir empfehlen, dass Ihr Verschlüsselungskontext die zu verschlüsselnden oder zu entschlüsselnden Daten beschreibt. Wenn Sie z. B. eine Datei verschlüsseln, können Sie einen Teil des Dateipfads als Verschlüsselungskontext verwenden.

```
"encryptionContext": {
    "department": "10103.0"
}
```

Wenn Sie beispielsweise Volumes und Snapshots verschlüsseln, die mit der Amazon Elastic Block Store (Amazon EBS)-CreateSnapshotOperation erstellt wurden, verwendet Amazon EBS die Volume-ID als Verschlüsselungskontextwert.

```
"encryptionContext": {
  "aws:ebs:id": "vol-abcde12345abc1234"
}
```

Sie können den Verschlüsselungskontext außerdem verwenden, um den Zugriff auf AWS KMS keys in Ihrem Konto detailliert festzulegen oder einzuschränken. Sie können den Verschlüsselungskontext als Einschränkung in Erteilungen und als Bedingung in Richtlinien verwenden.

Informationen zur Verwendung des Verschlüsselungskontexts zum Schutz der Integrität verschlüsselter Daten finden Sie im Beitrag So schützen Sie die Integrität Ihrer verschlüsselten Daten mithilfe von AWS Key Management Service und EncryptionContext im -AWSSicherheitsblog.

Weitere Informationen zum Verschlüsselungskontext.

Regeln für den Verschlüsselungskontext

AWS KMS erzwingt die folgenden Regeln für Verschlüsselungskontextschlüssel und -werte.

- Der Schlüssel und der Wert in einem Verschlüsselungskontextpaar müssen einfache Literalzeichenfolgen sein. Wenn Sie einen anderen Typ verwenden, z. B. eine Ganzzahl oder Gleitkommazahl, interpretiert AWS KMS ihn als Zeichenfolge.
- Die Schlüssel und Werte in einem Verschlüsselungskontext können Unicode-Zeichen enthalten. Wenn ein Verschlüsselungskontext Zeichen enthält, die in Schlüsselrichtlinien oder IAM-Richtlinien nicht zulässig sind, können Sie den Verschlüsselungskontext nicht in

Richtlinienbedingungsschlüsseln angeben, z. B. `kms:EncryptionContext:context-key` und `kms:EncryptionContextKeys`. Weitere Informationen zu Dokumentenregeln für Schlüsselrichtlinien finden Sie unter Schlüsselrichtlinienformat. Weitere Informationen zu Regeln für IAM-Richtliniendokumente finden Sie unter Anforderungen für den IAM-Namen im IAM Benutzerhandbuch.

Verschlüsselungskontext in Richtlinien

Der Verschlüsselungskontext wird hauptsächlich verwendet, um Integrität und Authentizität zu überprüfen. Sie können den Verschlüsselungskontext aber auch verwenden, um den Zugriff auf AWS KMS keys mit symmetrischer Verschlüsselung in Schlüsselrichtlinien und IAM-Richtlinien zu kontrollieren.

Die Bedingungsschlüssel kms:EncryptionContext: und kms:EncryptionContextKeys erlauben (oder verweigern) eine Berechtigung nur, wenn die Anforderung bestimmte Verschlüsselungskontextschlüssel oder Schlüssel-Wert-Paare enthält.

Die folgende Schlüsselrichtlinienanweisung erlaubt es beispielsweise der `RoleForExampleApp`-Rolle, den KMS-Schlüssel in `Decrypt`-Operationen zu verwenden. Sie verwendet den `kms:EncryptionContext:context-key`-Bedingungsschlüssel, um diese Berechtigung nur zu erlauben, wenn der Verschlüsselungskontext in der Anforderung ein `AppName:ExampleApp`-Verschlüsselungskontextpaar enthält.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Weitere Informationen zu diesen Bedingungskontextschlüsseln finden Sie unter Zustandstasten für AWS KMS.

Verschlüsselungskontext in Erteilungen

Wenn Sie eine Erteilung erstellen, können Sie Einschränkungen für Erteilungen setzten, die Bedingungen für die Erteilungsberechtigungen festlegen. AWS KMS unterstützt zwei Erteilungseinschränkungen, `EncryptionContextEquals` und `EncryptionContextSubset`, die beide den Verschlüsselungskontext in einer Anforderung für eine kryptografische Operation mit einbeziehen. Wenn Sie diese Erteilungseinschränkungen verwenden, sind die Berechtigungen in der Erteilung nur dann wirksam, wenn der Verschlüsselungskontext in der Anforderung für die kryptografische Operation die Anforderungen der Erteilungseinschränkungen erfüllt.

Sie können beispielsweise eine `EncryptionContextEquals` Erteilungseinschränkung zu einer Erteilung hinzufügen, die den GenerateDataKey Vorgang zulässt. Bei dieser Einschränkung erlaubt die Erteilung die Operation nur dann, wenn der Verschlüsselungskontext in der Anforderung mit dem Verschlüsselungskontext in der Erteilungseinschränkung übereinstimmen (inkl. Groß-/Kleinschreibung).

```
$ 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}
```

Eine Anforderung wie die folgende vom Empfänger-Prinzipal würde die `EncryptionContextEquals`-Einschränkung zulassen.

```
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --encryption-context Purpose=Test
```

Weitere Informationen über die Erteilungseinschränkungen finden Sie unter Verwenden von Erteilungs-Einschränkungen. Ausführliche Informationen zu Erteilungen finden Sie unter the section called “Gewährungen”.

Protokollieren des Verschlüsselungskontexts

AWS KMS verwendet AWS CloudTrail zum Protokollieren des Verschlüsselungskontext, sodass Sie bestimmen können, auf welche KMS-Schlüssel und Daten zugegriffen wurde. Das bedeutet, dass der

Protokolleintrag genau anzeigt, welcher KMS-Schlüssel verwendet wurde, um spezifische Daten, auf die der Verschlüsselungskontext im Protokolleintrag hinweist, zu verschlüsseln oder entschlüsseln.

> **Important**
>
> Da der Verschlüsselungskontext protokolliert wird, sollte er keine vertraulichen Informationen enthalten.

Speichern des Verschlüsselungskontexts

Um die Verwendung eines beliebigen Verschlüsselungskontextes beim Aufruf der Operationen `Decrypt` oder `ReEncrypt` zu vereinfachen, können Sie den Verschlüsselungskontext neben den verschlüsselten Daten speichern. Wir empfehlen, dass Sie nur so viel vom Verschlüsselungskontext speichern, dass Sie den vollständigen Verschlüsselungskontext erstellen können, wenn Sie ihn für die Ver- oder Entschlüsselung benötigen.

Wenn der Verschlüsselungskontext beispielsweise der vollständig qualifizierte Pfad zu einer Datei ist, speichern Sie nur einen Teil dieses Pfades mit dem verschlüsselten Dateiinhalt. Wenn Sie dann den vollständigen Verschlüsselungstext benötigen, können Sie ihn aus dem gespeicherten Fragment rekonstruieren. Wenn jemand die Datei verändert (sie beispielsweise umbenennt oder an einen anderen Ort verschiebt), ändert sich der Wert des Verschlüsselungskontextes und die Entschlüsselungsanforderung schlägt fehl.

Schlüsselrichtlinie

Wenn Sie einen KMS-Schlüssel erstellen, legen Sie fest, wer diesen KMS-Schlüssel verwenden und verwalten darf. Diese Berechtigungen werden in einem Dokument namens Schlüsselrichtlinie festgehalten. Sie können die Schlüsselrichtlinie verwenden, um Berechtigungen für einen kundenverwalteten KMS-Schlüssel jederzeit hinzuzufügen, zu entfernen oder zu ändern. Sie können die Schlüsselrichtlinie für einen Von AWS verwaltete Schlüssel jedoch nicht bearbeiten. Weitere Informationen finden Sie unter Wichtige Richtlinien in AWS KMS.

Gewährung

Eine Erteilung ist ein Richtlinien-Instrument, das es AWS-Prinzipalen erlaubt, AWS KMS keys in kryptografischen Operationen zu verwenden. Es kann ihnen auch erlauben, einen KMS-Schlüssel anzuzeigen (DescribeKey) und Erteilungen zu erstellen und zu verwalten. Bei der Autorisierung des Zugriffs auf einen KMS-Schlüssel werden Erteilungen zusammen mit Schlüsselrichtlinien und

IAM-Richtlinien berücksichtigt. Erteilungen werden häufig für temporäre Berechtigungen verwendet, da Sie eine erstellen, deren Berechtigungen verwenden und sie dann wieder löschen können, ohne Ihre Schlüsselrichtlinien oder IAM-Richtlinien zu ändern. Da Erteilungen sehr spezifisch sein können und einfach zu erstellen und zu widerrufen sind, werden sie häufig verwendet, um temporäre Berechtigungen oder detailliertere Berechtigungen bereitzustellen.

Ausführliche Informationen zu Erteilungen, einschließlich der Terminologie für Erteilungen, finden Sie unter Erteilungen in AWS KMS.

Prüfung der KMS-Schlüsselnutzung

Sie können verwendenAWS CloudTrail, um die Schlüsselnutzung zu überprüfen. CloudTrail erstellt Protokolldateien, die einen Verlauf von AWS API-Aufrufen und zugehörigen Ereignissen für Ihr Konto enthalten. Diese Protokolldateien enthalten alle AWS KMS-API-Anforderungen, die über die AWS-Managementkonsole, AWS-SDKs und Befehlszeilen-Tools erfolgen. Die Protokolldateien enthalten auch Anforderungen an AWS KMS, die AWS-Services in Ihrem Namen stellen. Sie können diese Protokolldateien verwenden, um wichtige Informationen zu finden, z. B. wann der KMS-Schlüssel verwendet wurde, die angeforderte Operation, die Identität des Anforderers und die Quell-IP-Adresse. Weitere Informationen finden Sie im Protokollierung mit AWS CloudTrail und dem AWS CloudTrail-Benutzerhandbuch.

Schlüsselverwaltungsinfrastruktur

Eine gängige Praxis in der Kryptographie besteht darin, Ver- und Entschlüsselungen mit einem öffentlich verfügbaren und von Kollegen überprüften Algorithmus wie AES (Advanced Encryption Standard) und einem geheimen Schlüssel vorzunehmen. Eines der größten Probleme bei der Kryptographie ist, dass es sehr schwierig ist, einen Schlüssel geheim zu halten. Dies ist in der Regel die Aufgabe einer Schlüsselverwaltungs-Infrastruktur (Key Management Infrastructure, KMI). AWS KMS betreibt die Schlüsselinfrastruktur für Sie. AWS KMS erstellt Ihre Stammschlüssel (AWS KMS keys genannt) und speichert sie auf sichere Weise. Weitere Informationen darüber, wie AWS KMS funktioniert, finden Sie im Whitepaper AWS Key Management Service – Kryptographische Details.

Schlüssel verwalten

Erstellen Sie zunächst einen AWS KMS key, um mit AWS KMS zu starten.

Die Themen in diesem Abschnitt beschreiben, wie Sie Ihren Basic-KMS-Schlüssel, einen KMS-Schlüssel zur symmetrischen Verschlüsselung, von der Erstellung bis zum Löschen verwalten. Sie umfassen Themen zum Bearbeiten und Anzeigen von Schlüsseln, Markieren von Schlüsseln, Aktivieren und Deaktivieren von Schlüsseln, Rotieren von Schlüsselmaterial und Verwenden von AWS-Tools und -Services zur Überwachung der Nutzung Ihrer KMS-Schlüssel. Sie enthalten auch Informationen zur Verwendung von AWS CloudFormation zum Erstellen und Verwalten Ihrer KMS-Schlüssel und eine Schlüsselstatusreferenz, die den erforderlichen Schlüsselstatus für jede AWS KMS-Produktion zeigt.

Weitere Informationen zum Erstellen, Verwenden und Verwalten anderer Arten von KMS-Schlüsseln finden Sie unter Schlüssel für spezielle Zwecke.

Themen

Erstellen von Schlüsseln

Sie können AWS KMS keys in der AWS Management Consoleoder mithilfe der -CreateKeyOperation oder einer AWS CloudFormation -Vorlage erstellen. Während dieses Prozesses wählen Sie den Typ

des KMS-Schlüssels, seine Regionalität (einzelne Region oder mehrere Regionen) und den Ursprung des Schlüsselmaterials aus (standardmäßig wird das Schlüsselmaterial von AWS KMS erstellt). Diese Eigenschaften können nicht geändert werden, nachdem der KMS-Schlüssel erstellt wurde. Sie legen außerdem die Schlüsselrichtlinie für den KMS-Schlüssel fest. Diese können Sie jederzeit ändern.

In diesem Thema wird erläutert, wie Sie den grundlegenden KMS-Schlüssel, einen KMS-Schlüssel mit symmetrischer Verschlüsselung, für eine einzelne Region mit Schlüsselmaterial aus AWS KMS erstellen. Sie können diesen KMS-Schlüssel verwenden, um Ihre Ressourcen in einer AWS-Service zu schützen. Ausführliche Informationen zu KMS-Schlüsseln mit symmetrischer Verschlüsselung finden Sie unter Schlüsselspezifikation SYMMMETRIC_DEFAULT. Hilfe beim Erstellen anderer Schlüsseltypen finden Sie unter Schlüssel für spezielle Zwecke.

Wenn Sie einen KMS-Schlüssel erstellen, um die Daten zu verschlüsseln, die Sie in einem AWS-Service speichern oder verwalten, erstellen Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung. AWS-Services, die in AWS KMS integriert sind, verwenden nur KMS-Schlüssel mit symmetrischer Verschlüsselung, um Ihre Daten zu verschlüsseln. Diese Services unterstützen keine Verschlüsselung mit asymmetrischen KMS-Schlüsseln. Informationen zur Entscheidung, welcher KMS-Schlüsseltyp erstellt werden soll, finden Sie unter Auswahl eines KMS-Schlüsseltyps.

> **Note**
>
> Symmetrische KMS-Schlüssel werden jetzt KMS-Schlüssel mit symmetrischer Verschlüsselung genannt. AWS KMS unterstützt zwei Arten symmetrischer KMS-Schlüssel, KMS-Schlüssel mit symmetrischer Verschlüsselung (der Standardtyp) und HMAC-KMS-Schlüssel, die auch symmetrische Schlüssel sind.

Wenn Sie einen KMS-Schlüssel in der AWS KMS-Konsole verwenden, müssen Sie ihm einen Alias (Anzeigename) geben. Die `CreateKey`-Produktion erstellt keinen Alias für den neuen KMS-Schlüssel. Um einen Alias für einen neuen oder vorhandenen KMS-Schlüssel zu erstellen, verwenden Sie die -CreateAliasOperation. Ausführliche Informationen zu Aliasen in AWS KMS finden Sie unter Verwenden von Aliassen.

In diesem Thema wird erläutert, wie Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung erstellen. In der folgenden Tabelle finden Sie Anleitungen zur Erstellung von KMS-Schlüsseln verschiedener Typen.

Anweisungen zum Erstellen eines KMS-Schlüssels

KMS-Schlüsseltyp	Anweisungen
Symmetrischer Verschlüsselungsschlüssel (SYMMETRIC_DEFAULT)	the section called “Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung”
Asymmetrischer Schlüssel	the section called “Erstellen asymmetrischer KMS-Schlüssel”
HMAC-Schlüssel	the section called “Erstellen von HMAC-Schlüsseln”
Schlüssel für mehrere Regionen (beliebiger Art)	the section called “Erstellen eines Primärschlüssels mit importiertem Schlüsselmaterial” the section called “Erstellen eines Replikatschlüssels mit importiertem Schlüsselmaterial”
Importiertes Schlüsselmaterial („Bring your own key – BYOK“)	the section called “Schritt 1: Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial”
AWS CloudHSM-Schlüsselspeicher	the section called “Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher”
Externer Schlüsselspeicher („Hold your own key – HYOK“)	the section called “Erstellen von KMS-Schlüsseln in einem externen Schlüsselspeicher”

Weitere Informationen:

- Um Datenschlüssel für die clientseitige Verschlüsselung zu erstellen, verwenden Sie die -GenerateDataKeyOperation.
- Informationen zum Erstellen eines asymmetrischen KMS-Schlüssels zur Verschlüsselung oder Signierung finden Sie unter Erstellen asymmetrischer KMS-Schlüssel.
- Informationen zum Erstellen eines HMAC-KMS-Schlüssels finden Sie unter Erstellen von HMAC-KMS-Schlüsseln.

- Informationen zum Erstellen eines KMS-Schlüssels mit importiertem Schlüsselmaterial („Bring Your Own Key“) finden Sie unter Importieren von Schlüsselmaterial Schritt 1: Erstellen eines AWS KMS key ohne Schlüsselmaterial.
- Weitere Informationen zur Erstellung eines multiregionalen Primär- oder Replikatschlüssels finden Sie unter Erstellen von multiregionalen Schlüsseln.
- Informationen zum Erstellen eines KMS-Schlüssels in einem benutzerdefinierten Schlüsselspeicher (Ursprung des Schlüsselmaterials ist ein benutzerdefinierter Schlüsselspeicher (CloudHSM)) finden Sie unter Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher.
- Informationen zur Verwendung einer -AWS CloudFormationVorlage zum Erstellen eines KMS-Schlüssels finden Sie AWS::KMS::Key unter im AWS CloudFormation -Benutzerhandbuch.
- Um festzustellen, ob ein bestehender KMS-Schlüssel symmetrisch oder asymmetrisch ist, lesen Sie unter Erkennen asymmetrischer KMS-Schlüssel nach.
- Um Ihre KMS-Schlüssel in Programmen oder in Befehlszeilenschnittstellen-Operationen zu verwenden, benötigen Sie eine Schlüssel-ID oder einen Schlüssel-ARN. Detaillierte Anweisungen finden Sie unter Finden der Schlüssel-ID und des Schlüssel-ARN.
- Weitere Informationen zu Kontingenten, die für KMS-Schlüssel gelten, finden Sie unter Kontingente.

Themen

Berechtigungen zum Erstellen von KMS-Schlüsseln

Um einen KMS-Schlüssel in der Konsole oder mithilfe der APIs zu erstellen, benötigen Sie die folgende Berechtigung in einer IAM-Richtlinie. Wenn möglich, verwenden Sie Bedingungsschlüssel, um die Berechtigungen einzuschränken. Sie können beispielsweise den Bedingungsschlüssel kms:KeySpec in einer IAM-Richtlinie verwenden, um Prinzipalen zu erlauben, nur symmetrische Verschlüsselungsschlüssel zu erstellen.

Ein Beispiel für eine IAM-Richtlinie für Prinzipale, die Schlüssel erstellen, finden Sie unter Einem Benutzer das Erstellen von KMS-Schlüsseln erlauben.

> **Note**
>
> Seien Sie vorsichtig, wenn Sie Prinzipalen die Berechtigung zum Verwalten von Tags und Aliasen erteilen. Wenn Sie einen Tag oder einen Alias ändern, wird dadurch die Berechtigung für den kundenverwalteten Schlüssel erteilt oder verweigert. Details hierzu finden Sie unter ABAC für AWS KMS.

- kms:CreateKey ist erforderlich.
- kms:CreateAlias ist erforderlich, um einen KMS-Schlüssel in der Konsole zu erstellen, in der für jeden neuen KMS-Schlüssel ein Alias erforderlich ist.
- kms:TagResource ist erforderlich, um beim Erstellen des KMS-Schlüssels Tags hinzuzufügen.
- iam:CreateServiceLinkedRole ist erforderlich, um multiregionale Primärschlüssel zu erstellen. Details hierzu finden Sie unter Steuern des Zugriffs auf multiregionale Schlüssel.

kms:PutKeyPolicy Die Berechtigung ist nicht erforderlich, um den KMS-Schlüssel zu erstellen. Die `kms:CreateKey`-Berechtigung enthält die Berechtigung zum Festlegen der ursprünglichen Schlüsselrichtlinie. Sie müssen diese Berechtigung jedoch der Schlüsselrichtlinie hinzufügen, während Sie den KMS-Schlüssel erstellen, um sicherzustellen, dass Sie den Zugriff auf den KMS-Schlüssel steuern können. Alternativ können Sie den BypassLockoutSafetyCheck Parameter verwenden, was nicht empfohlen wird.

KMS-Schlüssel gehören zu dem AWS-Konto, in dem sie erstellt wurden. Der IAM-Benutzer, der einen KMS-Schlüssel erstellt, gilt nicht als Schlüsselbesitzer und hat nicht automatisch die Berechtigung, den von ihm erstellten KMS-Schlüssel zu verwenden oder zu verwalten. Wie jeder andere Prinzipal muss der Schlüsselersteller eine Berechtigung über eine Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung erhalten. Prinzipale, die die `kms:CreateKey`-Berechtigung haben, können jedoch die ursprüngliche Schlüsselrichtlinie festlegen und sich selbst die Berechtigung zur Verwendung oder Verwaltung des Schlüssels erteilen.

Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung

Sie können KMS-Schlüssel in der AWS Management Console oder mit der AWS KMS-API erstellen.

In diesem Thema wird erläutert, wie Sie den grundlegenden KMS-Schlüssel, einen KMS-Schlüssel mit symmetrischer Verschlüsselung, für eine einzelne Region mit Schlüsselmaterial aus AWS KMS erstellen. Sie können diesen KMS-Schlüssel verwenden, um Ihre Ressourcen in einer AWS-Service

zu schützen. Hilfe beim Erstellen anderer Schlüsseltypen finden Sie unter Schlüssel für spezielle Zwecke.

Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung (Konsole)

Sie können die AWS Management Console zum Erstellen von AWS KMS keys (KMS-Schlüsseln) verwenden.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Um einen KMS-Schlüssel mit symmetrischer Verschlüsselung zu erstellen, wählen Sie für Key type (Schlüsseltyp) die Option Symmetric (Symmetrisch) aus.

   Weitere Informationen zum Erstellen eines asymmetrischen KMS-Schlüssels in der AWS KMS-Konsole finden Sie unter Erstellen asymmetrischer KMS-Schlüssel (Konsole).
6. Unter Key usage (Schlüsselverwendung) ist die Option Encrypt and decrypt (Verschlüsseln und Entschlüsseln) für Sie ausgewählt.

   Weitere Informationen wie Sie KMS-Schlüssel erstellen, die MAC-Codes generieren und überprüfen, finden Sie unter Erstellen von HMAC-KMS-Schlüsseln.
7. Wählen Sie Weiter aus.

   Weitere Hinweise zu den Advanced Options (Advanced Optionen) finden Sie unter Schlüssel für spezielle Zwecke.

8. Geben Sie einen Alias für den Replikatschlüssel ein. Der Aliasname darf nicht mit **`aws/`** beginnen. Das Präfix **`aws/`** ist von Amazon Web Services reserviert und steht für Von AWS verwaltete Schlüssel in Ihrem Konto.

   > **Note**
   >
   > Wenn Sie einen Alias hinzufügen, löschen oder aktualisieren, wird dadurch möglicherweise eine Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel.

   Ein Alias ist ein Anzeigename, den Sie verwenden können, um einen KMS-Schlüssel zu identifizieren. Wir empfehlen, dass Sie einen Alias wählen, der auf die Art von Daten, die Sie schützen möchten, oder die Anwendung, die Sie mit dem KMS-Schlüssel verwenden möchten, hindeutet.

   Zum Erstellen eines KMS-Schlüssels in der Konsole benötigen Sie Aliase AWS Management Console. Sie sind optional, wenn Sie die -CreateKeyOperation verwenden.

9. (Optional) Geben Sie eine Beschreibung für den KMS-Schlüssel ein.

   Sie können jetzt eine Beschreibung hinzufügen oder sie jederzeit aktualisieren, es sei denn, der Schlüsselstatus lautet `Pending Deletion` oder `Pending Replica Deletion`. Um die Beschreibung eines vorhandenen kundenverwalteten Schlüssels hinzuzufügen, zu ändern oder zu löschen, bearbeiten Sie die Beschreibung in der AWS Management Console oder verwenden Sie die -UpdateKeyDescriptionOperation.

10. (Optional) Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wählen Sie Add tag (Tag hinzufügen), wenn Sie mehr als ein Tag zum KMS-Schlüssel hinzufügen möchten.

    > **Note**
    >
    > Wenn Sie einen KMS-Schlüssel markieren oder entmarkieren, wird dadurch möglicherweise die Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.

Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

11. Wählen Sie Weiter aus.
12. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel verwalten können.

> **Note**
>
> Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung zum Verwalten des KMS-Schlüssels erteilen. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.
>
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

13. (Optional) Um zu verhindern, dass die ausgewählten IAM-Benutzer und -Rollen diesen KMS-Schlüssel löschen, deaktivieren Sie unten auf der Seite im Abschnitt Key deletion (Schlüssellöschung) das Kontrollkästchen Allow key administrators to delete this key (Administratoren erlauben, diesen Schlüssel zu löschen).
14. Wählen Sie Weiter aus.
15. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel für kryptographische Operationen verwenden können

> **Note**
>
> Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung erteilen, den KMS-Schlüssel in kryptografischen Operationen zu verwenden. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.

> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

16. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other AWS-Konten (Andere Konten) unten auf der Seite die Option Add another AWS-Konto (Weiteres Konto hinzufügen) und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

> **Note**
>
> Um auch Prinzipalen aus den externen Konten Zugriff auf den KMS-Schlüssel zu erlauben, müssen die Administratoren der externen Konten IAM-Richtlinien erstellen, die diese Berechtigungen bereitstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

17. Wählen Sie Weiter.
18. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
19. Wählen Sie Finish (fertigstellen) aus, um den KMS-Schlüssel zu erstellen.

Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung (AWS KMS-API)

Sie können die -CreateKeyOperation verwenden, um AWS KMS keys von allen Typen zu erstellen. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Der folgende Vorgang erstellt den am häufigsten verwendeten KMS-Schlüssel, einen symmetrischen Verschlüsselungsschlüssel in einer einzigen Region, der von AWS KMS generiert wird. Diese Produktion hat keine erforderlichen Parameter. Gegebenenfalls können Sie auch mit dem Parameter `Policy` eine Schlüsselrichtlinie angeben. Sie können die Schlüsselrichtlinie (PutKeyPolicy) ändern und optionale Elemente wie eine Beschreibung und Tags jederzeit hinzufügen. Darüber hinaus können Sie Asymmetrische Schlüssel, Schlüssel zu mehreren Regionen, Schlüssel mit Importiertes Schlüsselmaterial und Schlüssel in Custom Key Store erstellen.

Mit der -`CreateKey`Operation können Sie keinen Alias angeben, aber Sie können die -CreateAliasOperation verwenden, um einen Alias für Ihren neuen KMS-Schlüssel zu erstellen.

Es folgt ein Beispiel für einen Aufruf der Produktion `CreateKey` ohne Parameter. Dieser Befehl verwendet alle Standardwerte. Es erstellt einen KMS-Schlüssel mit symmetrischer Verschlüsselung zum Verschlüsseln und Entschlüsseln mit von AWS KMS generiertem Schlüsselmaterial.

```
$ 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"
        ],
    }
}
```

Wenn Sie keine Schlüsselrichtlinie für Ihren neuen KMS-Schlüssel angeben, unterscheidet sich die Standard-Schlüsselrichtlinie, die `CreateKey` verwendet, von der Standard-Schlüsselrichtlinie, die die Konsole anwendet, wenn Sie einen neuen KMS-Schlüssel erstellen.

Dieser Aufruf der -GetKeyPolicyOperation gibt beispielsweise die Schlüsselrichtlinie zurück, die `CreateKey` gilt. Sie gibt dem AWS-Konto Zugriff auf den KMS-Schlüssel und erlaubt es ihm, AWS Identity and Access Management (IAM)-Richtlinien für den KMS-Schlüssel zu erstellen. Detaillierte Informationen zu IAM-Richtlinien und Schlüsselrichtlinien für KMS-Schlüssel finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS

```
$ 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" : "*"
  } ]
}
```

Verwenden von Aliassen

Ein Alias ist ein Anzeigename für einen AWS KMS key. Mit einem Alias können Sie beispielsweise auf einen KMS-Schlüssel als `test-key`anstelle von `1234abcd-12ab-34cd-56ef-1234567890ab` verweisen.

Sie können einen Alias verwenden, um einen KMS-Schlüssel in der AWS KMSKonsole, in der -DescribeKeyOperation und in kryptografischen Operationen wie Encrypt und zu identifizierenGenerateDataKey. Darüber hinaus erleichtern Aliasse das Erkennen eines Von AWS verwalteter Schlüssel. Aliase für diese KMS-Schlüssel haben immer die Form `aws/<service-name>`. Beispiel: Der Alias für den Von AWS verwalteter Schlüssel für Amazon DynamoDB ist `aws/dynamodb`. Sie können ähnliche Alias-Standards für Ihre Projekte festlegen, z. B. den Namen eines Projekts oder einer Kategorie vorab für Ihre Aliasse.

Sie können den Zugriff auf KMS-Schlüssel auch anhand ihrer Aliasse zulassen und verweigern, ohne Richtlinien zu bearbeiten oder Erteilungen zu verwalten. Diese Funktion ist Teil der AWS KMS-Unterstützung für attributbasierte Zugriffssteuerung (ABAC). Details hierzu finden Sie unter Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel.

Ein Großteil der Möglichkeiten von Aliassen ergibt sich aus Ihrer Fähigkeit, den KMS-Schlüssel, der einem Alias zugeordnet ist, jederzeit zu ändern. Aliasse machen Ihren Code einfacher zu schreiben und zu verwalten. Angenommen, Sie verwenden einen Alias, um auf einen bestimmten KMS-Schlüssel zu verweisen, und Sie möchten den KMS-Schlüssel ändern. In diesem Fall ordnen Sie den Alias einfach einem anderen KMS-Schlüssel zu. Sie müssen keine Codeänderungen ausführen.

Darüber hinaus erleichtern Aliasse die Wiederverwendung des gleichen Codes in verschiedenen AWS-Regionen. Erstellen Sie Aliasse mit demselben Namen in mehreren Regionen und ordnen Sie jeden Alias einem KMS-Schlüssel in seiner Region zu. Wenn der Code in jeder Region ausgeführt wird, bezieht sich der Alias auf den zugeordneten KMS-Schlüssel in dieser Region. Ein Beispiel finden Sie unter Verwenden von Aliassen in Ihren Anwendungen.

Sie können einen Alias für einen KMS-Schlüssel in der -AWS KMSKonsole, mithilfe der CreateAlias-API oder mithilfe einer AWS CloudFormation -Vorlage erstellen.

Die AWS KMS-API bietet volle Kontrolle über Aliasse in jedem Konto und in jeder Region. Die API enthält Operationen zum Erstellen eines Alias (CreateAlias), Anzeigen von Aliasnamen und Alias-ARNs (ListAliases), Ändern des KMS-Schlüssels, der einem Alias zugeordnet ist (UpdateAlias) und Löschen eines Alias (DeleteAlias). Beispiele zur Erstellung und Verwaltung von Aliassen in verschiedenen Programmiersprachen finden Sie unter the section called “Arbeiten mit Aliasen”.

Die folgenden Ressourcen können Ihnen dabei helfen, mehr zu erfahren:

- Informationen über Schlüsselbezeichner, einschließlich Aliasse, finden Sie unter Schlüsselkennungen (KeyId).
- Hilfe bei der Verwendung einer AWS CloudFormation Vorlage zum Erstellen eines Alias für einen KMS-Schlüssel finden Sie unter AWS::KMS::Alias im AWS CloudFormation -Benutzerhandbuch.
- Hilfe zum Auffinden der Aliasse, die einem KMS-Schlüssel zugeordnet sind, finden Sie unter Suchen des Aliasnamens und des Alias-ARN
- Informationen zu Ressourcenkontingenten für Aliasse und Raten-Kontingente für API-Operationen im Zusammenhang mit Aliasnamen finden Sie unter Kontingente.
- Beispiele zur Erstellung und Verwaltung von Aliassen in verschiedenen Programmiersprachen finden Sie unter Arbeiten mit Aliasen.

Themen

- Verwenden von Aliassen in Ihren Anwendungen
- Steuern des Zugriffs auf Aliasse
- Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel
- Suchen von Aliassen in AWS CloudTrail-Protokollen

Über Aliasse

Erfahren Sie, wie Aliasse in AWS KMS funktionieren.

Ein Alias ist eine unabhängige AWS-Ressource

Ein Alias ist keine Eigenschaft eines KMS-Schlüssels. Die Aktionen, die Sie für den Alias ausführen, wirken sich nicht auf den zugeordneten KMS-Schlüssel aus. Sie können einen Alias für einen KMS-Schlüssel erstellen und dann den Alias aktualisieren, damit er einem anderen KMS-Schlüssel zugeordnet ist. Sie können den Alias sogar löschen, ohne dass sich auf den zugeordneten KMS-Schlüssel auswirkt. Wenn Sie jedoch einen KMS-Schlüssel löschen, werden alle Aliasse gelöscht, die diesem KMS-Schlüssel zugeordnet sind.

Wenn Sie einen Alias als Ressource in einer IAM-Richtlinie angeben, bezieht sich die Richtlinie auf den Alias und nicht auf den zugeordneten KMS-Schlüssel.

Jeder Alias hat zwei Formate

Beim Erstellen eines Aliasses geben Sie den Aliasnamen an. AWS KMS erstellt den Alias-ARN für Sie.

- Ein Alias-ARN ist ein Amazon-Ressourcenname (ARN), der den Alias eindeutig identifiziert.

```
# Alias ARN
arn:aws:kms:us-west-2:111122223333:alias/<alias-name>
```

- Der Aliasname muss im Konto und in der Region eindeutig sein. In der AWS KMS-API wird dem Aliasnamen immer das `alias/`-Präfix erteilt. Das Präfix ist in der AWS KMS-Konsole ausgelassen.

```
# Alias name
alias/<alias-name>
```

Aliase sind nicht geheim

Aliase können in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden. Geben Sie keine vertraulichen oder sensiblen Informationen in den Alias-Namen ein.

Jeder Alias ist jeweils einem KMS-Schlüssel zugeordnet

Der Alias und sein KMS-Schlüssel müssen sich im selben Konto und in derselben Region befinden.

Sie können einen Alias jedem kundenverwalteten Schlüssel im gleichen AWS-Konto und der gleichen Region zuordnen. Sie haben jedoch keine Berechtigung, einen Alias einem Von AWS verwalteter Schlüssel zuzuordnen.

Diese ListAliases Ausgabe zeigt beispielsweise, dass der Alias genau einem Ziel-KMS-Schlüssel zugeordnet ist, der durch die `test-key` -`TargetKeyId`Eigenschaft dargestellt wird.

```
{
     "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
}
```

Mehrere Aliasse können demselben KMS-Schlüssel zugeordnet werden

Beispielsweise können Sie die `test-key`- und `project-key`-Aliasse demselben KMS-Schlüssel zuordnen.

```
{
     "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
```

```
}
```

Ein Alias muss in einem Konto und einer Region eindeutig sein.

Zum Beispiel können Sie nur einen `test-key`-Alias in jedem Konto und in jeder Region haben. Aliasse beachten die Groß-/Kleinschreibung, aber Aliasse, die sich nur in ihrer Großschreibung unterscheiden, sind sehr fehleranfällig. Sie können keinen Aliasnamen ändern. Sie können den Alias jedoch löschen und einen neuen Alias mit dem gewünschten Namen erstellen.

Sie können jedoch einen Alias mit demselben Namen in verschiedenen Regionen erstellen.

Sie können beispielsweise einen `finance-key`-Alias In USA Ost (Nord-Virginia) und einen `finance-key`-Alias in Europa (Frankfurt) haben. Jeder Alias wäre einem KMS-Schlüssel in seiner Region zugeordnet. Wenn Ihr Code auf einen Aliasnamen wie `alias/finance-key` verweist, können Sie ihn in mehreren Regionen ausführen. In jeder Region wird ein anderer KMS-Schlüssel verwendet. Details hierzu finden Sie unter Verwenden von Aliassen in Ihren Anwendungen.

Sie können den KMS-Schlüssel ändern, der einem Alias zugeordnet ist.

Sie können die -UpdateAliasOperation verwenden, um einen Alias einem anderen KMS-Schlüssel zuzuordnen. Wenn beispielsweise der `finance-key`-Alias dem `1234abcd-12ab-34cd-56ef-1234567890ab`-KMS-Schlüssel zugeordnet ist, verwenden, können Sie ihn so aktualisieren, dass er dem `0987dcba-09fe-87dc-65ba-ab0987654321`-KMS-Schlüssel zugeordnet wird.

Der aktuelle KMS-Schlüssel und der neue KMS-Schlüssel müssen jedoch vom selben Typ sein (beide symmetrisch oder beide asymmetrisch oder beide HMAC) und die gleiche Schlüsselnutzung (ENCRYPT_DECRYPT oder SIGN_VERIFY oder GENERATE_VERIFY_MAC) aufweisen. Diese Beschränkung verhindert Fehler in Code, der Aliasse verwendet. Wenn Sie einen Alias einem anderen Schlüsseltyp zuordnen müssen und die Risiken verringert haben, können Sie den Alias löschen und neu erstellen.

Einige KMS-Schlüssel haben keine Aliasse

Wenn Sie einen KMS-Schlüssel in der AWS KMS-Konsole erstellen, müssen Sie ihm einen neuen Alias geben. Ein Alias ist jedoch nicht erforderlich, wenn Sie die -CreateKeyOperation verwenden, um einen KMS-Schlüssel zu erstellen. Außerdem können Sie die -UpdateAliasOperation verwenden, um den KMS-Schlüssel zu ändern, der einem Alias zugeordnet ist, und die -DeleteAliasOperation, um einen Alias zu löschen. Daher könnten manche KMS-Schlüssel mehrere Aliasse haben, während andere keinen haben.

AWS erstellt Aliasse in Ihrem Konto

AWS erstellt Aliasse in Ihrem Konto für Von AWS verwaltete Schlüssel. Diese Aliasse haben Namen des Formulars `alias/aws/<service-name>`, wie beispielsweise `alias/aws/s3`.

Manche AWS-Aliasse haben keinen KMS-Schlüssel. Diese vordefinierten Aliasse sind normalerweise einem Von AWS verwalteter Schlüssel zugeordnet, wenn Sie mit der Verwendung des Services beginnen.

Verwenden von Aliassen zum Identifizieren von KMS-Schlüsseln

Sie können einen Aliasnamen oder Alias-ARN verwenden, um einen KMS-Schlüssel in kryptografischen Operationen , DescribeKeyund zu identifizierenGetPublicKey. (Wenn sich der KMS-Schlüssel in einem anderen AWS-Konto befindet, müssen Sie seinen Schlüssel-ARN oder Alias-ARN verwenden.) Aliasse sind keine gültigen Bezeichner für KMS-Schlüssel in anderen AWS KMS-Operationen. Informationen zu den gültigen Schlüsselbezeichnern für jede AWS KMS-API-Operation finden Sie in den Beschreibungen der `KeyId`-Parameter in der AWS Key Management Service-API-Referenz.

Sie können keinen Aliasnamen oder Alias-ARN verwenden, um einen KMS-Schlüssel in einer IAM-Richtlinie zu identifizieren. Um den Zugriff auf einen KMS-Schlüssel basierend auf seinen Aliassen zu steuern, verwenden Sie die Bedingungsschlüssel kms:RequestAlias oder kms:ResourceAliases. Details hierzu finden Sie unter ABAC für AWS KMS.

Verwalten von Aliassen

Autorisierte Benutzer können Aliasse erstellen, anzeigen und löschen. Sie können Aliasse auch aktualisieren, wodurch ein vorhandenes Alias einem anderen KMS-Schlüssel zugeordnet wird.

Themen

- Erstellen eines Alias
- Anzeigen von Aliassen
- Aktualisieren von Aliassen
- Löschen eines Alias

Erstellen eines Alias

Sie können Aliasse in der AWS KMS-Konsole oder mithilfe von AWS KMS-API-Operationen erstellen.

Der Alias muss aus einer Zeichenfolge von 1–256 Zeichen bestehen. Er kann nur alphanumerische Zeichen (A-Z, a-z, 0-9), Bindestriche (-), Schrägstriche (/), Unterstriche (_) und Bindestriche (-) enthalten. Der Aliasname für einen kundenverwalteten Schlüssel kann nicht mit `alias/aws/`beginnen. Das Präfix `alias/aws/` ist für Von AWS verwalteter Schlüssel reserviert.

Sie können einen Alias für einen neuen KMS-Schlüssel oder einen vorhandenen KMS-Schlüssel erstellen. Sie können einen Alias hinzufügen, sodass ein bestimmter KMS-Schlüssel in einem Projekt oder einer Anwendung verwendet wird.

Erstellen eines Alias (Konsole)

Wenn Sie einen KMS-Schlüssel in der AWS KMS-Konsole erstellen, müssen Sie einen Alias für den neuen KMS-Schlüssel erstellen. Um einen Alias für einen vorhandenen KMS-Schlüssel zu erstellen, verwenden Sie die Aliasse-Registerkarte auf der Detailseite für den KMS-Schlüssel.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. Sie können keine Aliasse für Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel.
4. Wählen Sie in der Tabelle die Schlüssel-ID oder den Alias des KMS-Schlüssels. Wählen Sie dann auf der Detailseite für den KMS-Schlüssel die Aliasse-Registerkarte.

   Wenn ein KMS-Schlüssel über mehrere Aliasse verfügt, zeigt die Aliasse-Spalte in der Tabelle einen Alias und eine Aliasübersicht an, z. B. (+n mehr). Wenn Sie die Aliasübersicht auswählen, gelangen Sie direkt zur Aliases (Aliasse)-Registerkarte auf der Seite mit den KMS-Schlüsseldetails.
5. Wählen Sie auf der Aliasse-Registerkarte Create Alarm (Alarm erstellen). Geben Sie einen Aliasnamen ein und wählen Sie Create alias (Alias erstellen).

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

> **Note**
>
> Fügen Sie das Präfix `alias/` nicht hinzu. Die Konsole fügt es automatisch für Sie hinzu. Wenn Sie `alias/ExampleAlias` eingeben, wird der tatsächliche Aliasname `alias/alias/ExampleAlias` sein.

Erstellen eines Aliasses (AWS KMS-API)

Um einen Alias zu erstellen, verwenden Sie die -CreateAliasOperation. Im Gegensatz zum Erstellen von KMS-Schlüsseln in der Konsole erstellt die CreateKey Operation keinen Alias für einen neuen KMS-Schlüssel.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

Sie können die `CreateAlias`-Operation verwenden, um einen Alias für einen neuen KMS-Schlüssel ohne Alias zu erstellen. Sie können auch die `CreateAlias`-Operation verwenden, um einen Alias zu einem vorhandenen KMS-Schlüssel hinzuzufügen oder einen Alias neu zu erstellen, der versehentlich gelöscht wurde.

In den AWS KMS-API-Operationen muss jeder Aliasname mit `alias/` beginnen, gefolgt von einem Namen, z. B. `alias/ExampleAlias`. Ein Alias muss im Konto und der Region eindeutig sein. Verwenden Sie die Operation , um die Aliasnamen zu finden, die bereits verwendet werdenListAliases. Bei dem Aliasnamen wird zwischen Groß-/Kleinschreibung unterschieden.

Die `TargetKeyId` kann jeder kundenverwaltete Schlüssel in der gleichen AWS-Region sein. Verwenden Sie zum Identifizieren des KMS-Schlüssels seine Schlüssel-ID oder den Schlüssel-ARN. Sie können keinen anderen Alias verwenden.

Das folgende Beispiel erstellt den `example-key`-Alias und ordnet ihn dem angegebenen KMS-Schlüssel zu. Diese Beispiele verwenden die AWS Command Line Interface (AWS CLI). Beispiele in verschiedenen Programmiersprachen finden Sie unter Arbeiten mit Aliasen.

```
$ aws kms create-alias \
```

```
    --alias-name alias/example-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

`CreateAlias` gibt keine Ausgabe zurück. Um den neuen Alias anzuzeigen, verwenden Sie die `ListAliases`-Operation. Details hierzu finden Sie unter Anzeigen von Aliassen (AWS KMS-API).

Anzeigen von Aliassen

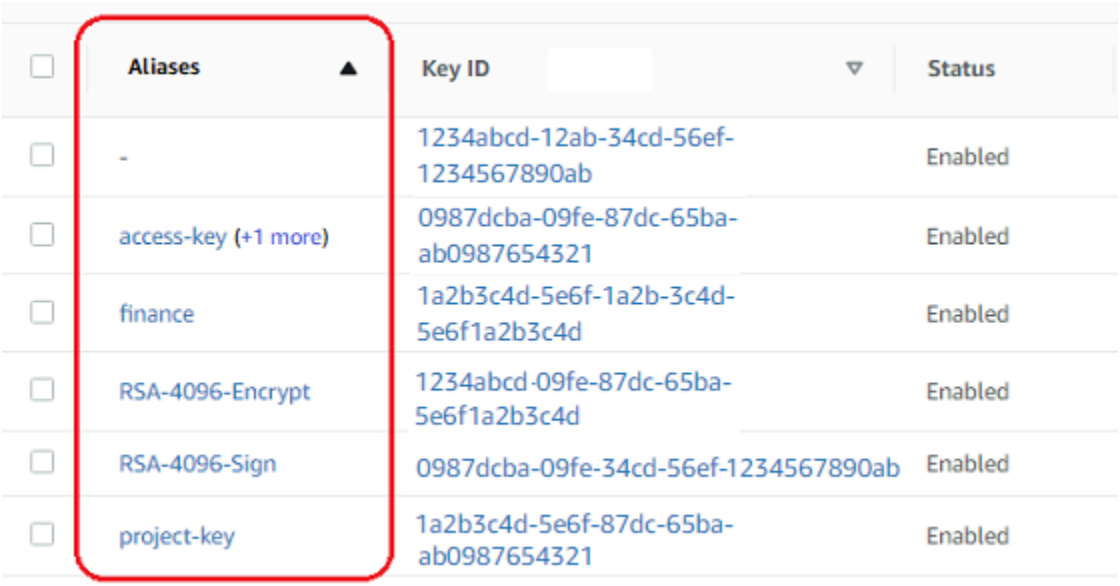
Mithilfe von Aliassen können Sie KMS-Schlüssel einfach in der AWS KMS-Konsole identifizieren. Sie können die Aliase für einen KMS-Schlüssel in der -AWS KMSKonsole oder mithilfe der -ListAliasesOperation anzeigen. Die -DescribeKeyOperation, die die Eigenschaften eines KMS-Schlüssels zurückgibt, enthält keine Aliase.

Anzeigen von Aliassen (Konsole)

Die Registerkarten Customer managed keys (kundenverwaltete Schlüssel) und Von AWS verwaltete Schlüssel in der AWS KMS-Konsole zeigen den Alias an, der jedem KMS-Schlüssel zugeordnet ist. Sie können auch KMS-Schlüssel nach ihren Aliassen suchen, sortieren und filtern.

Die folgende Abbildung der AWS KMS-Konsole zeigt die Aliasse auf der Seite Customer managed keys (kundenverwaltete Schlüssel) eines Beispielkontos. Wie in der Abbildung gezeigt, haben einige KMS-Schlüssel keinen Alias.

Wenn ein KMS-Schlüssel über mehrere Aliasse verfügt, zeigt die Aliasse-Spalte in der Tabelle einen Alias und eine Aliasübersicht an, z. B. (+n mehr). Die Aliasübersicht zeigt an, wie viele zusätzliche Aliasse dem KMS-Schlüssel zugeordnet sind, sowie Links zur Anzeige aller Aliasse für den KMS-Schlüssel auf der Aliasse-Registerkarte.

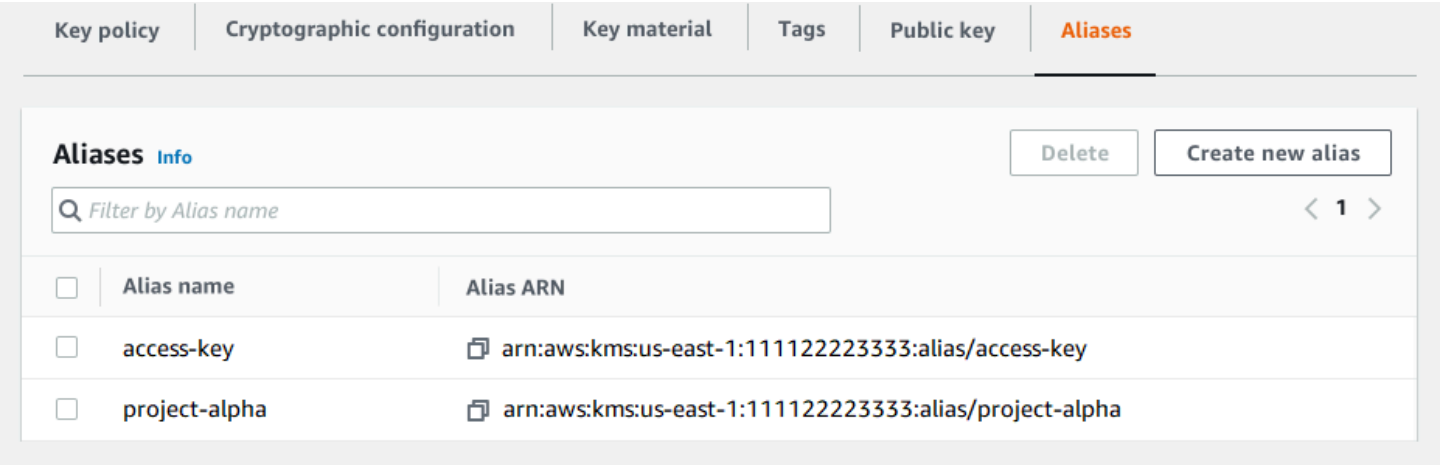


Die Aliasse-Registerkarte auf der Detailseite für jeden KMS-Schlüssel zeigt den Aliasnamen und den Alias-ARN aller Aliasse für den KMS-Schlüssel im AWS-Konto und der Region an. Sie können auch die Aliasse-Registerkarte verwenden, um Aliasse zu erstellen und Aliasse zu löschen.

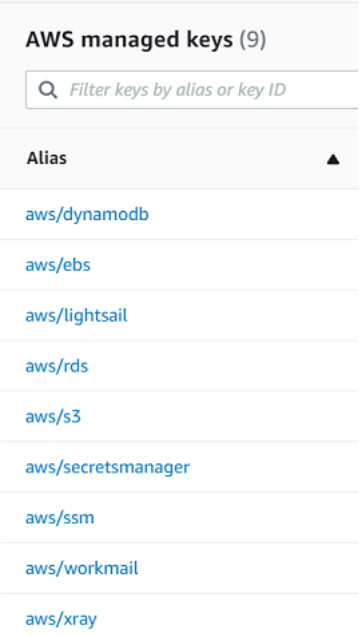
Um den Aliasnamen und den Alias-ARN aller Aliasse für den KMS-Schlüssel zu finden, verwenden Sie die Aliasse-Registerkarte.

- Um direkt zur Registerkarte Aliases (Aliase) zu gelangen, wählen Sie in der Spalte Aliases (Aliase) die Aliasübersicht aus (+n mehr). Eine Aliasübersicht wird nur angezeigt, wenn der KMS-Schlüssel mehr als einen Alias hat.
- Oder wählen Sie den Alias oder die Schlüssel-ID des KMS-Schlüssels aus (wodurch die Detailseite für den KMS-Schlüssel geöffnet wird) und wählen Sie dann die Aliase-Registerkarte. Die Registerkarten werden unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.

Die folgende Abbildung zeigt die Aliasse-Registerkarte für einen Beispiel-KMS-Schlüssel.



Sie können den Alias verwenden, um einen Von AWS verwalteter Schlüssel zu erkennen, wie in dieser Beispielseite für Von AWS verwaltete Schlüssel. Die Aliasse für Von AWS verwaltete Schlüssel haben immer das Format: `aws/<service-name>`. Beispiel: Der Alias für den Von AWS verwalteter Schlüssel für Amazon DynamoDB ist `aws/dynamodb`.



Anzeigen von Aliassen (AWS KMS-API)

Die ListAliases Operation gibt den Aliasnamen und den Alias-ARN der Aliasse im Konto und in der Region zurück. Die Ausgabe enthält Aliasse für Von AWS verwaltete Schlüssel und für kundenverwaltete Schlüssel. Die Aliasse für Von AWS verwaltete Schlüssel haben immer das Format `aws/<service-name>`, z. B. `aws/dynamodb`.

Die Antwort kann auch Aliase ohne das Feld `TargetKeyId` enthalten. Dies sind vordefinierte Aliasse, die von AWS erstellt, aber noch keinem KMS-Schlüssel zugeordnet wurden.

```
$ 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
        }
    ]
}
```

Wenn Sie nur die Aliasse auflisten möchten, die einem bestimmten KMS-Schlüssel zugeordnet sind, verwenden Sie den optionalen `KeyId`-Parameter der `ListAliases`-Operation. Der `KeyId`-Parameter übernimmt die Schlüssel-ID oder den Schüssel-ARN des KMS-Schlüssels.

In diesem Beispiel werden alle Aliase abgerufen, die dem `0987dcba-09fe-87dc-65ba-ab0987654321`KMS-Schlüssel zugeordnet sind

```
$ 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
        }
    ]
}
```

Der `KeyId`-Parameter akzeptiert keine Platzhalterzeichen, aber Sie können die Funktionen Ihrer Programmiersprache verwenden, um die Antwort zu filtern.

Der folgende AWS CLI-Befehl ruft beispielsweise nur die Aliasse für Von AWS verwaltete Schlüssel ab.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

Der folgende Befehl ruft nur den `access-key`-Alias ab. Bei dem Aliasnamen wird zwischen Groß- und Kleinschreibung unterschieden.

```
$ 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"
    }
]
```

Aktualisieren von Aliassen

Da es sich bei einem Alias um eine unabhängige Ressource handelt, können Sie den KMS-Schlüssel ändern, der einem Alias zugeordnet ist. Wenn der Alias beispielsweise einem KMS`test-key`-

Schlüssel zugeordnet ist, können Sie die -UpdateAliasOperation verwenden, um ihn einem anderen KMS-Schlüssel zuzuordnen. Dies ist eine von mehreren Möglichkeiten, einen KMS-Schlüssel manuell zu drehen, ohne das Schlüsselmaterial zu ändern. Sie können auch einen KMS-Schlüssel aktualisieren, sodass eine Anwendung, die einen bestimmten KMS-Schlüssel für neue Ressourcen verwendet hat, jetzt einen anderen KMS-Schlüssel verwendet.

Sie können keinen Alias in der AWS KMS-Konsole aktualisieren. Sie können außerdem nicht `UpdateAlias` (oder eine andere Operation) verwenden, einen Aliasnamen zu ändern. Wenn Sie einen Aliasnamen ändern möchten, löschen Sie den aktuellen Alias und erstellen Sie einen neuen Alias für den KMS-Schlüssel.

Wenn Sie einen Alias aktualisieren, müssen der aktuelle KMS-Schlüssel und der neue KMS-Schlüssel vom selben Typ sein (beide symmetrisch oder asymmetrisch oder HMAC). Sie müssen auch die gleiche Schlüsselverwendung haben (`ENCRYPT_DECRYPT` oder `SIGN_VERIFY` oder GENERATE_VERIFY_MAC). Diese Beschränkung verhindert kryptografische Fehler in Code, der Aliasse verwendet.

Das folgende Beispiel beginnt mit der -ListAliasesOperation, um zu zeigen, dass der `test-key` Alias derzeit dem KMS-Schlüssel zugeordnet ist`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
        }
    ]
}
```

Als Nächstes ändert die `UpdateAlias`-Operation den KMS-Schlüssel, der dem `test-key`-Alias zugeordnet ist, in den KMS-Schlüssel `0987dcba-09fe-87dc-65ba-ab0987654321`. Sie müssen nicht den aktuell zugeordneten KMS-Schlüssel angeben, sondern nur den neuen („Ziel“)-KMS-Schlüssel. Bei dem Aliasnamen wird zwischen Groß-/Kleinschreibung unterschieden.

```
$ aws kms update-alias --alias-name 'alias/test-key' --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321
```

Um zu überprüfen, ob der Alias jetzt dem Ziel-KMS-Schlüssel zugeordnet ist, verwenden Sie die `ListAliases`-Operation erneut. Dieser AWS CLI-Befehl verwendet den `--query`-Parameter, um nur den `test-key`-Alias abzurufen. Die `TargetKeyId`- und `LastUpdatedDate`-Felder werden aktualisiert.

```
$ 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
    }
]
```

Löschen eines Alias

Sie können einen Alias in der -AWS KMSKonsole oder mithilfe der -DeleteAliasOperation löschen. Bevor Sie einen Alias löschen, stellen Sie sicher, dass er nicht verwendet wird. Obwohl das Löschen eines Aliasses keinen Einfluss auf den zugeordneten KMS-Schlüssel hat, kann es zu Problemen für jede Anwendung führen, die den Alias verwendet. Wenn Sie einen Alias versehentlich löschen, können Sie einen neuen Alias mit demselben Namen erstellen und diesen mit demselben oder einem anderen KMS-Schlüssel verknüpfen.

Wenn Sie einen KMS-Schlüssel löschen, werden alle Aliasse gelöscht, die diesem KMS-Schlüssel zugeordnet sind.

Löschen von Aliassen (Konsole)

Um einen Alias in der AWS KMS-Konsole zu löschen, verwenden Sie die Aliasse-Registerkarte auf der Detailseite für den KMS-Schlüssel. Sie können mehrere Aliase für einen KMS-Schlüssel gleichzeitig löschen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. Sie können keine Aliasse für Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel.

4. Wählen Sie in der Tabelle die Schlüssel-ID oder den Alias des KMS-Schlüssels. Wählen Sie dann auf der Detailseite für den KMS-Schlüssel die Aliasse-Registerkarte.

   Wenn ein KMS-Schlüssel über mehrere Aliasse verfügt, zeigt die Aliasse-Spalte in der Tabelle einen Alias und eine Aliasübersicht an, z. B. (+n mehr). Wenn Sie die Aliasübersicht auswählen, gelangen Sie direkt zur Aliases (Aliasse)-Registerkarte auf der Seite mit den KMS-Schlüsseldetails.

5. Aktivieren Sie auf der Aliasse-Registerkarte das Kontrollkästchen neben den Aliassen, die Sie löschen möchten. Wählen Sie dann Löschen.

Löschen eines Aliasses (AWS KMS-API)

Um einen Alias zu löschen, verwenden Sie die -DeleteAliasOperation. Diese Operation löscht jeweils einen Alias. Beim Aliasnamen wird zwischen Groß-/Kleinschreibung unterschieden und der Aliasname muss mit dem `alias/`-Präfix beginnen.

Der folgenden Befehl löscht beispielsweise den `test-key`-Alias. Dieser Befehl gibt keine Ausgabe zurück.

```
$ aws kms delete-alias --alias-name alias/test-key
```

Um zu überprüfen, ob der Alias gelöscht wurde, verwenden Sie die -ListAliasesOperation. Dieser Befehl verwendet den `--query`-Parameter in der AWS CLI, um nur den `test-key`-Alias abzurufen. Die leeren Klammern in der Antwort weisen darauf hin, dass die `ListAliases`-Antwort keinen `test-key`-Alias enthält. Um die Klammern zu entfernen, verwenden Sie den `--output text`-Parameter und -Wert.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[]
```

Verwenden von Aliassen in Ihren Anwendungen

Sie können einen Alias verwenden, um einen KMS-Schlüssel im Anwendungscode darzustellen. Der `KeyId` Parameter in AWS KMS kryptografischen Operationen , DescribeKeyund GetPublicKey akzeptiert einen Aliasnamen oder Alias-ARN.

Der folgende `GenerateDataKey`-Befehl verwendet beispielsweise einen Aliasnamen (`alias/finance`), um einen KMS-Schlüssel zu identifizieren. Der Aliasname ist der Wert des `KeyId`-Parameters.

```
$ aws kms generate-data-key --key-id alias/finance --key-spec AES_256
```

Wenn sich der KMS-Schlüssel in einem anderen AWS-Konto befindet, müssen Sie einen Schlüssel-ARN oder Alias-ARN in diesen Operationen verwenden.) Beachten Sie bei der Verwendung eines Alias-ARN, dass der Alias für einen KMS-Schlüssel in dem Konto definiert ist, das den KMS-Schlüssel besitzt und sich in jeder Region unterscheiden kann. Hilfestellung beim Suchen des Alias-ARN finden Sie unter Suchen des Aliasnamens und des Alias-ARN.

Der folgende `GenerateDataKey`-Befehl verwendet beispielsweise einen KMS-Schlüssel, der sich nicht im Konto des Anrufers befindet. Der `ExampleAlias`-Alias ist dem KMS-Schlüssel im angegebenen Konto und in der Region zugeordnet.

```
$ aws kms generate-data-key --key-id arn:aws:kms:us-west-2:444455556666:alias/
ExampleAlias --key-spec AES_256
```

Eine der mächtigsten Verwendungen von Aliasen ist in Anwendungen, die in mehreren AWS-Regionen ausgeführt werden. Beispielsweise könnten Sie über eine globale Anwendung verfügen, die eine RSA asymmetrischen KMS-Schlüssel zur Signatur und Verifizierung verwendet.

- In USA West (Oregon) (us-west-2) sollten Sie `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` verwenden.
- In Europa (Frankfurt) (eu-central-1) sollten Sie `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321` verwenden.
- In Asien-Pazifik (Singapur) (ap-southeast-1) sollten Sie `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d` verwenden.

Sie können in jeder Region eine andere Version Ihrer Anwendung erstellen oder ein Wörterbuch oder eine Switch-Anweisung verwenden, um den richtigen KMS-Schlüssel für jede Region auszuwählen. Es ist jedoch viel einfacher, in jeder Region einen Alias mit demselben Aliasnamen zu erstellen. Bei dem Aliasnamen wird zwischen Groß-/Kleinschreibung unterschieden.

```
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
```

Verwenden Sie dann den Alias in Ihrem Code. Wenn Ihr Code in jeder Region ausgeführt wird, verweist der Alias auf den zugehörigen KMS-Schlüssel in dieser Region. Beispielsweise ruft dieser Code die Sign-Operation mit einem Aliasnamen auf.

```
aws kms sign --key-id alias/new-app \
    --message $message \
    --message-type RAW \
    --signing-algorithm RSASSA_PSS_SHA_384
```

Es besteht jedoch das Risiko, dass der Alias gelöscht oder aktualisiert wird, um einem anderen KMS-Schlüssel zugeordnet zu werden. In diesem Fall schlagen die Versuche der Anwendung, Signaturen mit dem Aliasnamen zu überprüfen, fehl, und Sie müssen den Alias möglicherweise neu erstellen oder aktualisieren.

Um dieses Risiko zu verringern, sollten Sie den Prinzipalen die Berechtigung geben, die Aliase zu verwalten, die Sie in Ihrer Anwendung verwenden. Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.

Es gibt mehrere andere Lösungen für Anwendungen, die Daten in mehreren AWS-Regionen verschlüsseln, einschließlich des AWS Encryption SDK.

Steuern des Zugriffs auf Aliasse

Wenn Sie einen Alias erstellen oder ändern, wirkt sich das auf den Alias und den zugehörigen KMS-Schlüssel aus. Daher müssen Prinzipale, die Aliasse verwalten, über die Berechtigung verfügen, die Alias-Operation für den Alias und alle betroffenen KMS-Schlüssel aufzurufen. Sie können diese Berechtigungen mithilfe von Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen erteilen.

> **Note**
>
> Seien Sie vorsichtig, wenn Sie Prinzipalen die Berechtigung zum Verwalten von Tags und Aliassen erteilen. Wenn Sie eine Markierung oder einen Alias ändern, wird dadurch die Berechtigung für den kundenverwalteten Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel.

Weitere Hinweise zum Steuern des Zugriffs auf alle AWS KMS-Operationen finden Sie unter Berechtigungsreferenz.

Berechtigungen zum Erstellen und Verwalten von Aliasen funktionieren wie folgt.

kms:CreateAlias

Um einen Alias zu erstellen, benötigt der Prinzipal die folgenden Berechtigungen sowohl für den Alias als auch für den zugeordneten KMS-Schlüssel.

- `kms:CreateAlias` für den Alias. Geben Sie diese Berechtigung in einer IAM-Richtlinie an, die dem Prinzipal zugeordnet ist, der den Alias erstellen darf.

  Die folgende Beispiel-Richtlinienanweisung gibt einen bestimmten Alias in einem `Resource`-Element an. Sie können jedoch mehrere Alias-ARNs auflisten oder ein Aliasmuster wie "test*" angeben. Sie können auch einen `Resource`-Wert von `"*"` angeben, um dem Prinzipal zu erlauben, einen Alias im Konto und der Region zu erstellen. Die Berechtigung zum Erstellen eines Aliasses kann auch in einer `kms:Create*`-Berechtigung für alle Ressourcen in einem Konto und einer Region enthalten sein.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- `kms:CreateAlias` für den KMS-Schlüssel. Diese Berechtigung muss in einer Schlüsselrichtlinie oder in einer IAM-Richtlinie bereitgestellt werden, die von der Schlüsselrichtlinie delegiert wird.

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

Sie können Bedingungsschlüssel verwenden, um die KMS-Schlüssel einzuschränken, die Sie einem Alias zuordnen können. Sie können beispielsweise den kms:KeySpec-Bedingungsschlüssel verwenden, um dem Prinzipal zu erlauben, Aliase nur für asymmetrische KMS-Schlüssel zu erstellen. Eine vollständige Liste der Bedingungsschlüssel, mit denen Sie die `kms:CreateAlias`-Berechtigung für KMS-Schlüsselressourcen einschränken können, finden Sie unter AWS KMS Berechtigungen.

kms:ListAliases

Um Aliasse im Konto und in der Region aufzulisten, muss der Prinzipal über `kms:ListAliases`-Berechtigung in einer IAM-Richtlinie verfügen. Da diese Richtlinie nicht mit einem bestimmten KMS-Schlüssel oder Alias-Ressource verwandt ist, muss der Wert des Ressourcenelements in der Richtlinie `"*"` sein.

Beispielsweise gibt die folgende IAM-Richtlinienanweisung dem Prinzipal die Berechtigung, alle KMS-Schlüssel und Aliasse im Konto und in der Region aufzulisten.

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

```
  }
}
```

kms:UpdateAlias

Um den KMS-Schlüssel zu ändern, der einem Alias zugeordnet ist, benötigt der Prinzipal drei Berechtigungselemente: eines für den Alias, eines für den aktuellen KMS-Schlüssel und eines für den neuen KMS-Schlüssel.

Angenommen, Sie möchten z. B. den `test-key`-Alias aus dem KMS-Schlüssel mit Schlüssel-ID 1234abcd-12ab-34cd-56ef-1234567890ab zum KMS-Schlüssel mit Schlüssel-ID 0987dcba-09fe-87dc-65ba-ab0987654321 ändern. Fügen Sie in diesem Fall ähnliche Richtlinienanweisungen wie in den Beispielen in diesem Abschnitt ein.

- `kms:UpdateAlias` für den Alias. Sie stellen diese Berechtigung in einer IAM-Richtlinie zur Verfügung, die dem Prinzipal angefügt ist. Die folgende IAM-Richtlinie gibt einen bestimmten Alias an. Sie können jedoch mehrere Alias-ARNs auflisten oder ein Aliasmuster wie `"test*"` angeben. Sie können auch einen `Resource`-Wert von `"*"` angeben, um dem Prinzipal zu erlauben, einen Alias im Konto und der Region zu aktualisieren.

  ```
  {
    "Sid": "IAMPolicyForAnAlias",
    "Effect": "Allow",
    "Action": [
      "kms:UpdateAlias",
      "kms:ListAliases",
      "kms:ListKeys"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
  }
  ```

- `kms:UpdateAlias` für den KMS-Schlüssel, der derzeit dem Alias zugeordnet ist. Diese Berechtigung muss in einer Schlüsselrichtlinie oder in einer IAM-Richtlinie bereitgestellt werden, die von der Schlüsselrichtlinie delegiert wird.

  ```
  {
    "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` für den KMS-Schlüssel, den die Operation dem Alias zuordnet. Diese Berechtigung muss in einer Schlüsselrichtlinie oder in einer IAM-Richtlinie bereitgestellt werden, die von der Schlüsselrichtlinie delegiert wird.

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

Mit Hilfe von Bedingungsschlüsseln können Sie einen oder beide KMS-Schlüssel in einer `UpdateAlias`-Operation begrenzen. Sie können beispielsweise einen kms:ResourceAliases-Bedingungsschlüssel verwenden, damit der Prinzipal Aliase nur aktualisieren kann, wenn der Ziel-KMS-Schlüssel bereits über einen bestimmten Alias verfügt. Eine vollständige Liste der Bedingungsschlüssel, mit denen Sie die `kms:UpdateAlias`-Berechtigung für KMS-Schlüsselressourcen einschränken können, finden Sie unter AWS KMS Berechtigungen.

kms:DeleteAlias

Um einen Alias zu löschen, benötigt der Prinzipal die Berechtigung für den Alias und für den zugeordneten KMS-Schlüssel.

Wie immer sollten Sie Vorsicht walten lassen, wenn Sie Prinzipalen die Berechtigung zum Löschen einer Ressource erteilen. Das Löschen eines Aliasses hat keine Auswirkungen auf den zugehörigen KMS-Schlüssel. Es kann zwar zu einem Fehler in einer Anwendung führen, die auf dem Alias beruht, aber wenn Sie versehentlich einen Alias löschen, können Sie ihn neu erstellen.

- `kms:DeleteAlias` für den Alias. Sie stellen diese Berechtigung in einer IAM-Richtlinie zur Verfügung, die dem Prinzipal angefügt ist, der den Alias löschen darf.

Die folgende Beispiel-Richtlinienanweisung gibt einen bestimmten Alias in einem `Resource`-Element an. Sie können jedoch mehrere Alias-ARNs auflisten oder ein Aliasmuster angeben, z. B. `"test*"`. Sie können auch einen `Resource`-Wert von `"*"` angeben, damit der Prinzipal einen Alias im Konto und der Region löschen darf.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- `kms:DeleteAlias` für den zugeordneten KMS-Schlüssel. Diese Berechtigung muss in einer Schlüsselrichtlinie oder in einer IAM-Richtlinie bereitgestellt werden, die von der Schlüsselrichtlinie delegiert wird.

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

Begrenzen von Alias-Berechtigungen

Sie können Bedingungsschlüssel verwenden, um Alias-Berechtigungen einzuschränken, wenn es sich bei der Ressource um einen KMS-Schlüssel handelt. Die folgende IAM-Richtlinie erlaubt beispielsweise Alias-Operationen für KMS-Schlüssel in einem bestimmten Konto und einer

bestimmten Region. Es verwendet jedoch den kms:KeyOrigin-Bedingungsschlüssel, um die Berechtigungen auf KMS-Schlüssel mit Schlüsselmaterial von weiter einzuschränkenAWS KMS.

Eine vollständige Liste der Bedingungsschlüssel, mit denen Sie Alias-Berechtigungen für eine KMS-Schlüsselressource einschränken können, finden Sie unter AWS KMS Berechtigungen.

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

Sie können keine Bedingungsschlüssel in einer Richtlinienanweisung verwenden, bei der die Ressource ein Alias ist. Um die Aliasse einzuschränken, die ein Prinzipal verwalten kann, verwenden Sie den Wert des `Resource`-Elements der IAM-Richtlinienanweisung, die den Zugriff auf den Alias steuert. Mit den folgenden Richtlinienanweisungen kann der Prinzipal beispielsweise einen Alias im AWS-Konto und der Region erstellen, aktualisieren oder löschen, es sei denn, der Alias beginnt mit `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*"
}
```

Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel

Verwenden Sie diesen Bedingungsschlüssel, um den Zugriff auf einen KMS-Schlüssel anhand der Aliassen, die dem KMS-Schlüssel zugeordnet sind, zu steuern. Verwenden Sie dazu die Bedingungsschlüssel kms:RequestAlias und kms:ResourceAliases. Diese Funktion ist Teil der AWS KMS-Unterstützung für attributbasierte Zugriffssteuerung (ABAC).

Der `kms:RequestAlias`-Bedingungsschlüssel erlaubt oder verweigert den Zugriff auf einen KMS-Schlüssel basierend auf dem Alias in einer Anforderung. Der `kms:ResourceAliases`-Bedingungsschlüssel erlaubt oder verweigert den Zugriff auf einen KMS-Schlüssel basierend auf dem Alias, der mit dem KMS-Schlüssel verknüpft ist.

Diese Funktionen erlauben es Ihnen nicht, einen KMS-Schlüssel mithilfe eines Aliasses im `resource`-Element einer Richtlinienanweisung zu identifizieren. Wenn ein Alias der Wert eines`resource`-Element ist, gilt die Richtlinie für die Alias-Ressource und nicht für einen KMS-Schlüssel, der mit ihr verknüpft sein könnte.

> **Note**
>
> Es kann bis zu fünf Minuten dauern, bis Tag- und Alias-Änderungen Auswirkungen auf die KMS-Schlüsselautorisierung haben. Letzte Änderungen sind möglicherweise in API-Operationen sichtbar, bevor sie sich auf die Autorisierung auswirken.

Beachten Sie Folgendes, wenn Sie Aliasse verwenden, um den Zugriff auf KMS-Schlüssel zu steuern:

- Verwenden Sie Aliasse, um beim Zugriff die bewährte Methode der geringsten Berechtigung zu befolgen. Geben Sie IAM-Prinzipalen nur die Berechtigungen, die sie für die KMS-Schlüssel benötigen, die sie verwenden oder verwalten müssen. Verwenden Sie beispielsweise Aliase, um die KMS-Schlüssel zu identifizieren, die für ein Projekt verwendet werden. Geben Sie dann dem Projektteam die Berechtigung, nur KMS-Schlüssel mit den Projekt-Aliassen zu verwenden.

- Seien Sie vorsichtig, wenn Sie Prinzipalen die `kms:CreateAlias`-, `kms:UpdateAlias`- oder `kms:DeleteAlias`-Berechtigung erteilen, mit denen sie Aliasse hinzufügen, bearbeiten und löschen können. Wenn Sie Aliasse verwenden, um den Zugriff auf KMS-Schlüssel zu steuern, kann das Ändern eines Aliasses Prinzipalen die Berechtigung zur Verwendung von KMS-Schlüsseln erteilen, die andernfalls nicht über die Berechtigung verfügen. Es kann auch den Zugriff auf KMS-Schlüssel verweigern, die andere Prinzipale für ihre Aufträge benötigen.
- Überprüfen Sie die Prinzipale in Ihrem AWS-Konto, die derzeit über die Berechtigung verfügen, Aliasse zu verwalten und passen Sie ggf. die Berechtigungen an. Schlüsseladministratoren, die nicht über die Berechtigung zum Ändern von Schlüsselrichtlinien oder zum Erstellen von Erteilungen verfügen, können den Zugriff auf KMS-Schlüssel steuern, wenn sie über die Berechtigung zum Verwalten von Aliassen verfügen.

  Zum Beispiel enthält die Standard-Schlüsselrichtlinie für Schlüsseladministratoren der Konsole die `kms:CreateAlias`- `kms:DeleteAlias`- und `kms:UpdateAlias`-Berechtigung. IAM-Richtlinien geben möglicherweise dem Alias Berechtigungen für alle KMS-Schlüssel in Ihrem AWS-Konto. Die von AWSKeyManagementServicePowerUser verwaltete Richtlinie erlaubt es beispielsweise Prinzipalen, Aliase für alle KMS-Schlüssel zu erstellen, zu löschen und aufzulisten, sie jedoch nicht zu aktualisieren.
- Bevor Sie eine Richtlinie festlegen, die von einem Alias abhängt, überprüfen Sie die Aliasse auf den KMS-Schlüsseln in Ihrem AWS-Konto. Stellen Sie sicher, dass Ihre Richtlinie nur für die Aliase gilt, die Sie einschließen möchten. Verwenden Sie CloudTrail Protokolle und CloudWatch Alarme, um Sie auf Aliasänderungen aufmerksam zu machen, die sich auf den Zugriff auf Ihre KMS-Schlüssel auswirken könnten. Außerdem enthält die ListAliases Antwort das Erstellungsdatum und das Datum der letzten Aktualisierung für jeden Alias.
- Die Alias-Richtlinienbedingungen verwenden Musterabgleich; sie sind nicht an eine bestimmte Instance eines Aliasses gebunden. Eine Richtlinie, die Alias-basierte Bedingungsschlüssel verwendet, wirkt sich auf alle neuen und vorhandenen Aliasse aus, die dem Muster entsprechen. Wenn Sie einen Alias löschen und neu erstellen, der einer Richtlinienbedingung entspricht, gilt die Bedingung für den neuen Alias, genau wie für den alten Alias.

Der `kms:RequestAlias`-Bedingungsschlüssel basiert auf dem Alias, der explizit in einer Operations-Anforderung angegeben ist. Der `kms:ResourceAliases`-Bedingungsschlüssel hängt von den Aliassen ab, die einem KMS-Schlüssel zugeordnet sind, auch wenn sie nicht in der Anforderung angezeigt werden.

kms:RequestAlias

Erlauben oder verweigern Sie den Zugriff auf einen KMS-Schlüssel basierend auf dem Alias, der den KMS-Schlüssel in einer Anforderung identifiziert. Sie können den Bedingungsschlüssel kms:RequestAlias in einer Schlüsselrichtlinie oder IAM-Richtlinie verwenden. Sie gilt für Operationen, die einen Alias verwenden, um einen KMS-Schlüssel in einer Anforderung zu identifizieren, nämlich kryptografische Operationen , DescribeKeyund GetPublicKey. Sie ist nicht für Alias-Operationen wie CreateAlias oder gültigDeleteAlias.

Geben Sie im Bedingungsschlüssel einen Aliasnamen oder ein Aliasnamen-Muster an. Sie können keinen Alias-ARN angeben.

Die folgende Schlüsselrichtlinienanweisung erlaubt es beispielsweise Prinzipalen, die angegebenen Operationen auf dem KMS-Schlüssel zu verwenden. Die Berechtigung ist nur wirksam, wenn die Anforderung einen Alias verwendet, der `alpha` enthält, um den KMS-Schlüssel zu identifizieren.

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

Die folgende Beispielanforderung von einem autorisierten Prinzipal würde die Bedingung erfüllen. Eine Anforderung, die eine Schlüssel-ID, einen Schüssel-ARN oder einen anderen Alias verwendet, würde die Bedingung nicht erfüllen, selbst wenn diese Werte denselben KMS-Schlüssel identifizieren.

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-
alpha"
```

kms:ResourceAliases

Erlauben oder verweigern Sie den Zugriff auf einen KMS-Schlüssel anhand der Aliasse, die dem KMS-Schlüssel zugeordnet sind, selbst wenn der Alias nicht in einer Anforderung verwendet wird. Mit dem kms:ResourceAliases-Bedingungsschlüssel können Sie einen Alias oder ein Aliasmuster angeben, z. B. `alias/test*`, sodass Sie ihn in einer IAM-Richtlinie verwenden können, um den Zugriff auf mehrere KMS-Schlüssel in derselben Region zu steuern. Er ist für jede AWS KMS-Operation gültig, die einen KMS-Schlüssel verwendet.

Mit der folgenden IAM-Richtlinie können die Prinzipale beispielsweise die automatische Schlüsseldrehung auf den KMS-Schlüsseln in zwei AWS-Konten verwalten. Die Berechtigung gilt jedoch nur für KMS-Schlüssel, die Aliassen zugeordnet sind, die mit `restricted` beginnen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:EnableKeyRotation",
        "kms:DisableKeyRotation",
        "kms:GetKeyRotationStatus"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/*",
        "arn:aws:kms:*:444455556666:key/*"
      ],
      "Condition": {
        "ForAnyValue:StringLike": {
          "kms:ResourceAliases": "alias/restricted*"
        }
      }
    }
  ]
}
```

Die `kms:ResourceAliases`-Bedingung ist eine Bedingung der Ressource, nicht die Anforderung. Daher kann eine Anforderung, die den Alias nicht angibt, weiterhin die Bedingung erfüllen.

Die folgende Beispielanforderung, die einen übereinstimmenden Alias angibt, erfüllt die Bedingung.

```
$ aws kms enable-key-rotation --key-id "alias/restricted-project"
```

Die folgende Beispielanforderung erfüllt jedoch auch die Bedingung, vorausgesetzt, dass der angegebene KMS-Schlüssel über einen Alias verfügt, der mit `restricted` beginnt, auch wenn dieser Alias nicht in der Anforderung verwendet wird.

```
$ aws kms enable-key-rotation --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Suchen von Aliassen in AWS CloudTrail-Protokollen

Sie können einen Alias verwenden, um einen AWS KMS key in einer AWS KMS-API-Operation darzustellen. Wenn Sie dies tun, werden der Alias und der Schlüssel-ARN des KMS-Schlüssels im AWS CloudTrail-Protokolleintrag für das Ereignis aufgezeichnet. Der Alias erscheint im `requestParameters`-Feld. Der Schlüssel-ARN erscheint im `resources`-Feld. Dies gilt auch, wenn ein AWS-Service einen Von AWS verwalteter Schlüssel In Ihrem Konto verwendet.

Die folgende GenerateDataKey Anforderung verwendet beispielsweise den Alias `project-key` , um einen KMS-Schlüssel darzustellen.

```
$ aws kms generate-data-key --key-id alias/project-key --key-spec AES_256
```

Wenn diese Anforderung im CloudTrail Protokoll aufgezeichnet wird, enthält der Protokolleintrag sowohl den Alias als auch den Schlüssel-ARN des tatsächlich verwendeten KMS-Schlüssels.

```
{
    "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"
}
```

Weitere Informationen zum Protokollieren von AWS KMS Operationen in - CloudTrail Protokollen finden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

Anzeigen von Schlüsseln

Sie können AWS Management Console oder die AWS Key Management Service (AWS KMS)-API verwenden, um AWS KMS keys in jedem Konto und jeder Region anzuzeigen, einschließlich der von Ihnen verwalteten KMS-Schlüssel und der KMS-Schlüssel, die von AWS verwaltet werden.

Themen

- KMS-Schlüssel in der Konsole anzeigen
- Anzeigen von KMS-Schlüssel mit der API
- Anzeigen der kryptografischen Konfiguration von KMS-Schlüsseln
- Finden der Schlüssel-ID und des Schlüssel-ARN
- Suchen des Aliasnamens und des Alias-ARN

KMS-Schlüssel in der Konsole anzeigen

Im AWS Management Console können Sie Listen Ihrer KMS-Schlüssel im Konto und in der Region sowie Details zu jedem KMS-Schlüssel anzeigen.

> **Note**
>
> Die AWS KMS-Konsole zeigt die KMS-Schlüssel an, zu denen Sie Berechtigung zur Anzeige in Ihrem Konto und Ihrer Region haben. KMS-Schlüssel in anderen AWS-Konten werden nicht in der Konsole angezeigt, auch wenn Sie die Berechtigung haben, sie anzuzeigen, zu verwalten und zu verwenden. Um KMS-Schlüssel in anderen Konten anzuzeigen, verwenden Sie die -DescribeKeyOperation.

Themen

Navigieren zu den Schlüsseltabellen

Die AWS KMS keys in jedem Konto und jeder Region werden in Tabellen angezeigt. Es gibt separate Tabellen für die von Ihnen erstellten KMS-Schlüssel und die KMS-Schlüssel, die AWS-Services für Sie erstellen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus. Weitere Informationen zu den verschiedenen KMS-Schlüsseltypen finden Sie unter AWS KMS keys.

> **Tip**
>
> Zum Anzeigen der Von AWS verwaltete Schlüssel, die keinen Alias haben, verwenden Sie die Seite Customer managed keys (Vom Kunden verwaltete Schlüssel).
> Die AWS KMS-Konsole zeigt auch die benutzerdefinierten Schlüsselspeicher im Konto und in der Region an. KMS-Schlüssel, die Sie in benutzerdefinierten Schlüsselspeichern erstellen, werden auf der Seite Customer managed keys (kundenverwaltete Schlüssel) angezeigt. Hinweise zu benutzerdefinierten Schlüsselspeichern finden Sie unter Benutzerdefinierte Schlüsselspeicher.

Navigieren zu Schlüsseldetails

Es gibt eine Detailseite für jeden AWS KMS key in dem Konto und der Region. Auf der Detailseite wird die allgemeine Konfiguration für den KMS-Schlüssel angezeigt, einschließlich Registerkarten, mit denen autorisierte Benutzer die kryptografische Konfiguration und Schlüsselrichtlinie für den Schlüssel anzeigen und verwalten können. Je nach Schlüsseltyp enthält die Detailseite auch die Registerkarten Aliases (Aliasse), Key material (Schlüsselmaterial), Key rotation (Schlüsseldrehung), Public key (Öffentlicher Schlüssel), Regionality (Regionalität) und Tags.

So navigieren Sie zur Detailseite für einen KMS-Schlüssel.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus. Weitere Informationen zu den verschiedenen KMS-Schlüsseltypen finden Sie unter AWS KMS key.
4. Um die Seite mit den Schlüsseldetails zu öffnen, wählen Sie in der Schlüsseltabelle die Schlüssel-ID oder den Alias des KMS-Schlüssels aus.

   Wenn der KMS-Schlüssel mehrere Aliase enthält, wird eine Aliasübersicht (+n mehr) neben dem Namen des Aliases angezeigt. Wenn Sie die Aliasübersicht auswählen, gelangen Sie direkt zur Aliases (Aliase)-Registerkarte auf der Seite mit den Schlüsseldetails.

Sortieren und Filtern Ihrer KMS-Schlüssel

Um das Auffinden Ihrer KMS-Schlüssel in der Konsole zu erleichtern, können Sie die Schlüsseltabellen sortieren und filtern.

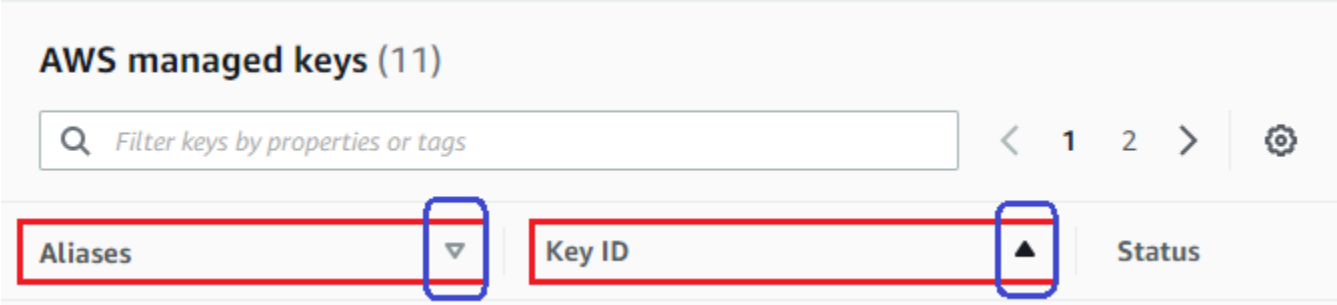
Sortierung

Sie können kundenverwaltete KMS-Schlüssel in aufsteigender oder absteigender Reihenfolge nach ihren Spaltenwerten sortieren. Diese Funktion sortiert alle KMS-Schlüssel in der Tabelle, auch wenn sie nicht auf der aktuellen Tabellenseite angezeigt werden.

Sortierbare Spalten werden durch einen Pfeil neben dem Spaltennamen gekennzeichnet. Auf der Seite Von AWS verwaltete Schlüssel können Sie nach Aliases (Aliase) oder Key ID (Schlüssel-ID) sortieren. Auf der Seite Customer managed keys (kundenverwaltete Schlüssel) können Sie nach Alias, Schlüssel-ID oder Schlüsseltyp sortieren.

Um in aufsteigender Reihenfolge zu sortieren, wählen Sie die Spaltenüberschrift aus, bis der Pfeil nach oben zeigt. Um in absteigender Reihenfolge zu sortieren, wählen Sie die Spaltenüberschrift, bis der Pfeil nach unten zeigt. Es kann jeweils nur nach einer Spalte sortiert werden.

Beispielsweise können Sie KMS-Schlüssel in aufsteigender Reihenfolge nach Schlüssel-ID sortieren, anstatt standardmäßig nach Alias.



Wenn Sie KMS-Schlüssel auf der Seite Customer managed keys (kundenverwaltete Schlüssel) in aufsteigender Reihenfolge nach Schlüsseltyp sortieren, werden alle asymmetrischen Schlüssel vor allen symmetrischen Schlüsseln angezeigt.

Filter

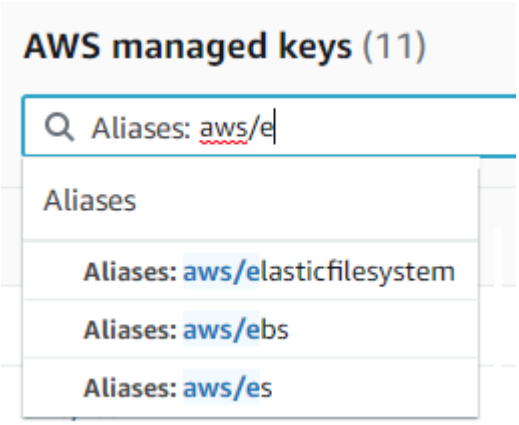
Sie können KMS-Schlüssel nach ihren Eigenschaftswerten oder Tags filtern. Der Filter gilt für alle KMS-Schlüssel in der Tabelle, auch wenn sie nicht auf der aktuellen Tabellenseite angezeigt werden. Bei dem Filter wird die Groß-/Kleinschreibung nicht berücksichtigt.

Filterbare Eigenschaften werden im Filterfeld aufgeführt. Auf der Seite Von AWS verwaltete Schlüssel können Sie nach Alias und Schlüssel-ID filtern. Auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) können Sie nach Alias, Schlüssel-ID und Schlüsseltyp-Eigenschaften sowie nach Tags filtern.

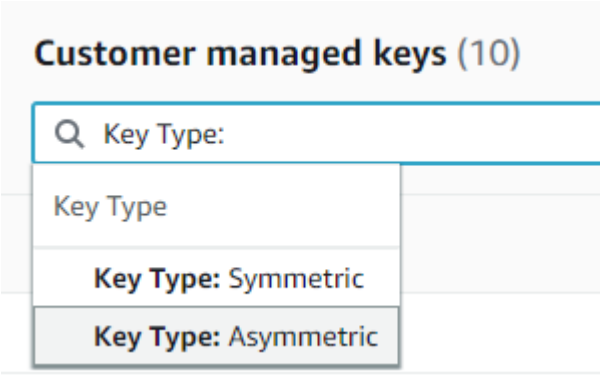
- Auf der Seite Von AWS verwaltete Schlüssel können Sie nach Alias und Schlüssel-ID filtern.
- Auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) können Sie nach Tags filtern, oder nach dem Alias, der Schlüssel-ID, des Schlüsseltyps oder den Regionalität-Eigenschaften.

Um nach einem Eigenschaftswert zu filtern, wählen Sie den Filter und den Eigenschaftsnamen aus und wählen Sie dann aus der Liste der tatsächlichen Eigenschaftswerte aus. Um nach einem Tag zu filtern, wählen Sie den Tag-Schlüssel aus, und wählen Sie dann aus der Liste der tatsächlichen Tag-Werte. Nachdem Sie eine Eigenschaft oder einen Tag-Schlüssel ausgewählt haben, können Sie den Eigenschaftswert oder den Tag-Wert auch ganz oder teilweise eingeben. Sie sehen eine Vorschau der Ergebnisse, bevor Sie Ihre Wahl treffen.

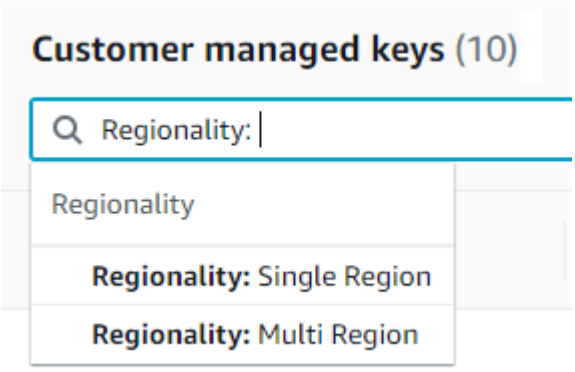
Um beispielsweise KMS-Schlüssel mit einem Aliasnamen anzuzeigen, der `aws/e` enthält, wählen Sie das Filterfeld und danach Alias aus, geben Sie `aws/e` ein und drücken Sie dann `Enter` oder `Return`, um den Filter hinzuzufügen.



Um nur asymmetrische KMS-Schlüssel auf der Seite Customer managed keys (kundenverwaltete Schlüssel) anzuzeigen, klicken Sie auf das Filterfeld, wählen Sie Schlüsseltyp und dann Schlüsseltyp: Asymmetrisch aus. Die Option Asymmetrisch wird nur angezeigt, wenn asymmetrische KMS-Schlüssel in der Tabelle vorhanden sind. Weitere Informationen zum Identifizieren von asymmetrischen KMS-Schlüsseln finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.



Um nur multiregionale Schlüssel anzuzeigen, wählen Sie auf der Seite Customer managed keys (kundenverwaltete Schlüssel) im Filterfeld Regionality (Regionalität) aus, und dann Regionality: Multi-Region. Die Multi-Region-Option wird nur angezeigt, wenn multiregionale Schlüssel in der Tabelle vorhanden sind. Weitere Informationen zum Identifizieren von multiregionalen Schlüsseln finden Sie unter Anzeigen von multiregionalen Schlüsseln.

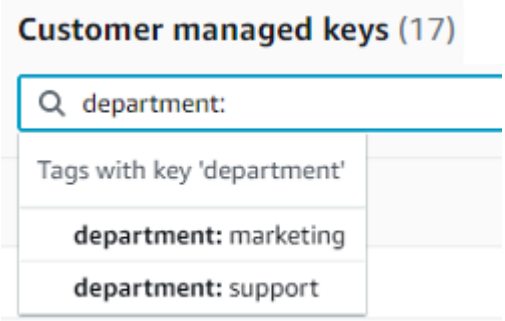


Die Tag-Filterung ist etwas anders. Um nur KMS-Schlüssel mit einem bestimmten Tag anzuzeigen, wählen Sie das Filterfeld aus, wählen Sie den Tag-Schlüssel aus, und wählen Sie dann einen der tatsächlichen Tag-Werte aus. Sie können den Tag-Wert auch ganz oder teilweise eingeben.

Es werden in der Tabelle alle KMS-Schlüssel mit dem ausgewählten Tag angezeigt. Das Tag wird jedoch nicht angezeigt. Um das Tag anzuzeigen, wählen Sie die Schlüssel-ID oder den Alias des KMS-Schlüssels aus und auf der Detailseite die Tags-Registerkarte. Die Registerkarte wird unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.

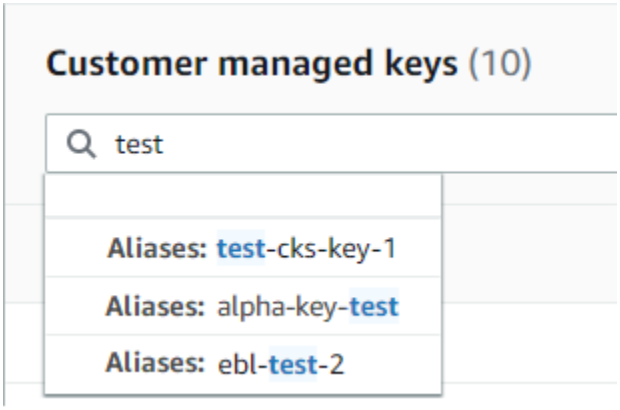
Dieser Filter erfordert sowohl den Tag-Schlüssel als auch den Tag-Wert. Es wird keine KMS-Schlüssel finden, indem nur der Tag-Schlüssel oder nur dessen Wert eingegeben wird.
Um Tags nach dem gesamten oder einem Teil des Tag-Schlüssels oder -Werts zu filtern, verwenden Sie die -ListResourceTagsOperation, um getaggte KMS-Schlüssel abzurufen, und

verwenden Sie dann die Filterfunktionen Ihrer Programmiersprache. Ein Beispiel finden Sie unter ListResourceTags: Abrufen der Tags für KMS-Schlüssel.

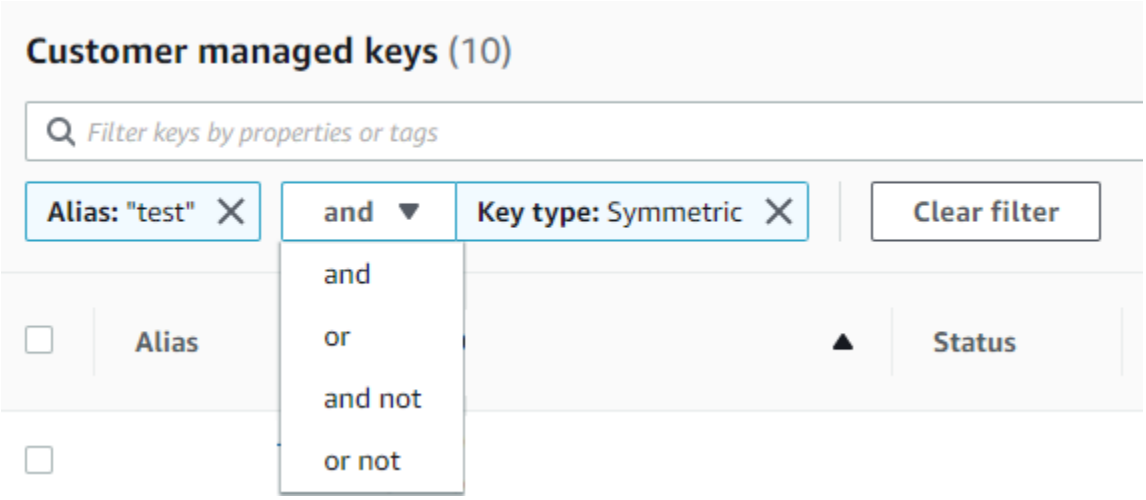


Um nach Text zu suchen, geben Sie im Filterfeld einen Alias, eine Schlüssel-ID, einen Schlüsseltyp oder einen Tag-Schlüssel ganz oder teilweise ein. (Nachdem Sie den Tag-Schlüssel ausgewählt haben, können Sie nach einem Tag-Wert suchen.) Sie sehen eine Vorschau der Ergebnisse, bevor Sie Ihre Wahl treffen.

Um beispielsweise KMS-Schlüssel mit `test` in ihren Tag-Schlüsseln oder filterbaren Eigenschaften anzuzeigen, geben Sie `test` in das Filterfeld ein. Die Vorschau zeigt die KMS-Schlüssel, die der Filter auswählen wird. In diesem Fall wird `test` nur in der Alias-Eigenschaft angezeigt.



Sie können mehrere Filter gleichzeitig verwenden. Wenn Sie zusätzliche Filter hinzufügen, können Sie auch einen logischen Operator auswählen.

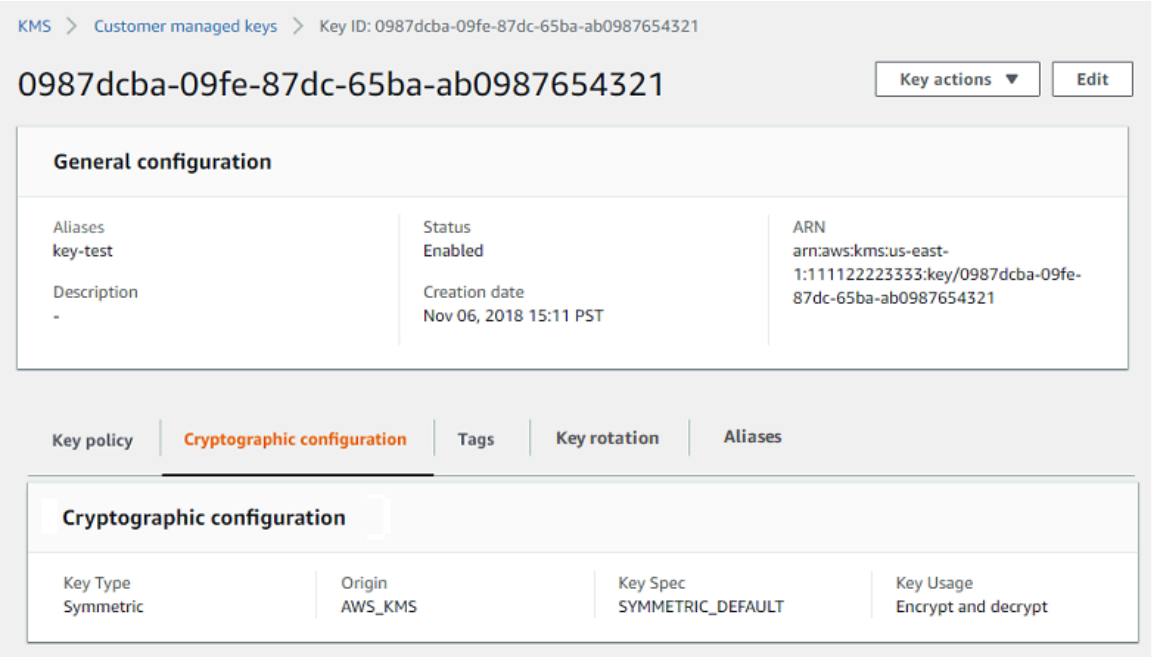


Anzeigen von KMS-Schlüsseldetails

Die Detailseite für jeden KMS-Schlüssel zeigt die Eigenschaften des KMS-Schlüssels an. Sie unterscheidet sich geringfügig für die verschiedenen Arten von KMS-Schlüsseln.

Um detaillierte Informationen zu einem KMS-Schlüssel anzuzeigen, wählen Sie auf der Seite Von AWS verwaltete Schlüssel oder Customer managed keys (Vom Kunden verwaltete Schlüssel) den Alias oder die Schlüssel-ID des KMS-Schlüssels aus.

Die Detailseite eines KMS-Schlüssels enthält den Abschnitt General Configuration (allgemeine Konfiguration), der die grundlegenden Eigenschaften des KMS-Schlüssels anzeigt. Sie enthält auch Registerkarten, auf denen Sie Eigenschaften des KMS-Schlüssels anzeigen und bearbeiten können, z. B. Schlüsselrichtlinie, kryptografische Konfiguration, Tags, Schlüsselmaterial (für KMS-Schlüssel mit importiertem Schlüsselmaterial), Schlüsseldrehung (für KMS-Schlüssel mit symmetrischer Verschlüsselung), Regionalität (für multiregionale Schlüssel) und öffentlicher Schlüssel (für asymmetrische KMS-Schlüssel).



In der folgenden Liste werden die Felder in der Detailanzeige beschrieben, einschließlich der Felder in den Registerkarten. Einige dieser Felder sind auch als Spalten in der Tabellenanzeige verfügbar.

Aliasnamen

Wo: Registerkarte für Aliase

Ein Anzeigename für den KMS-Schlüssel Ein Alias ist ein Anzeigename, den Sie verwenden können, um den KMS-Schlüssel in der Konsole und in einigen AWS KMS-APIs zu identifizieren. Details hierzu finden Sie unter Verwenden von Aliassen.

Auf der Registerkarte Aliases (Aliase) werden alle Aliase angezeigt, die dem KMS-Schlüssel in dem AWS-Konto und der Region zugeordnet sind.

ARN

Wo: Abschnitt über allgemeine Konfiguration

Der Amazon-Ressourcenname (ARN) eines KMS-Schlüssels. Dieser Wert identifiziert den KMS-Schlüssel eindeutig. Sie können damit den KMS-Schlüssel in AWS KMS-API-Operationen identifizieren.

Verbindungsstatus

Gibt an, ob ein benutzerdefinierter Schlüsselspeicher mit seinem Unterstützungsschlüsselspeicher verbunden ist. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellt wird.

Informationen zu den Werten in diesem Feld finden Sie unter ConnectionState in der APIAWS KMS-Referenz zu .

Erstellungsdatum

Wo: Abschnitt über allgemeine Konfiguration

Datum und Uhrzeit, wann der KMS-Schlüssel erstellt wurde. Dieser Wert wird in Ortszeit für das Gerät angezeigt. Die Zeitzone hängt nicht von der Region ab.

Anders als bei Ablauf bezieht sich die Erstellung nur auf den KMS-Schlüssel, nicht auf sein Schlüsselmaterial.

CloudHSM-Cluster-ID

Wo: Registerkarte für kryptografische Konfiguration

Die Cluster-ID des AWS CloudHSM-Clusters, der das Schlüsselmaterial für den KMS-Schlüssel enthält. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellt wird.

Wenn Sie die CloudHSM-Cluster-ID auswählen, wird die Cluster-Seite in der AWS CloudHSM-Konsole geöffnet.

ID des benutzerdefinierten Schlüsselspeichers

Wo: Registerkarte für kryptografische Konfiguration

Die ID des benutzerdefinierten Schlüsselspeichers, der den KMS-Schlüssel enthält. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellt wird.

Wenn Sie die ID des benutzerdefinierten Schlüsselspeichers auswählen, wird die Seite Custom key stores (benutzerdefinierte Schlüsselspeicher) in der AWS KMS-Konsole geöffnet.

Name des benutzerdefinierten Schlüsselspeichers

Wo: Registerkarte für kryptografische Konfiguration

Der Name des benutzerdefinierten Schlüsselspeichers, der den KMS-Schlüssel enthält. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellt wird.

Typ des benutzerdefinierten Schlüsselspeichers

Wo: Registerkarte für kryptografische Konfiguration

Gibt an, ob der benutzerdefinierte Schlüsselspeicher ein AWS CloudHSMSchlüsselspeicher oder ein externer Schlüsselspeicher ist. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellt wird.

Beschreibung

Wo: Abschnitt über allgemeine Konfiguration

Eine kurze, optionale Beschreibung des KMS-Schlüssels, die Sie schreiben und bearbeiten können. Um die Beschreibung eines kundenverwalteten Schlüssels hinzuzufügen oder zu aktualisieren, wählen Sie über General Configuration (allgemeine Konfiguration) Edit (Bearbeiten) aus.

Verschlüsselungsalgorithmen

Wo: Registerkarte für kryptografische Konfiguration

Listet die Verschlüsselungsalgorithmen auf, die mit dem KMS-Schlüssel in AWS KMS verwendet werden können. Dieses Feld wird nur angezeigt, wenn Key type (Schlüsseltyp) auf Asymmetric (Asymmetrisch) und Key usage (Schlüsselnutzung) auf Encrypt and decrypt (Verschlüsseln und Entschlüsseln) eingestellt ist. Informationen zu den von AWS KMS unterstützten Verschlüsselungsalgorithmen finden Sie unter Schlüsselspezifikation SYMMMETRIC_DEFAULT und RSA-Schlüsselspezifikationen für Verschlüsselung und Entschlüsselung.

Ablaufdatum

Wo: Registerkarte für Schlüsselmaterial

Datum und Uhrzeit des Zeitpunkts, an dem das Schlüsselmaterial für den KMS-Schlüssel abläuft. Dieses Feld wird nur bei KMS-Schlüsseln mit importiertem Schlüsselmaterial angezeigt, das heißt, wenn der Ursprung extern ist und der KMS-Schlüssel Schlüsselmaterial enthält, das abläuft.

ID des externen Schlüssels

Wo: Registerkarte für kryptografische Konfiguration

Die ID des externen Schlüssels, der einem KMS-Schlüssel in einem externen Schlüsselspeicher zugeordnet ist. Dieses Feld wird nur für KMS-Schlüssel in einem externen Schlüsselspeicher angezeigt.

Status des externen Schlüssels

Wo: Registerkarte für kryptografische Konfiguration

Der letzte Status, den der externe Schlüsselspeicher-Proxy für den externen Schlüssel gemeldet hat, der dem KMS-Schlüssel zugeordnet ist. Dieses Feld wird nur für KMS-Schlüssel in einem externen Schlüsselspeicher angezeigt.

Nutzung des externen Schlüssels

Wo: Registerkarte für kryptografische Konfiguration

Die kryptografischen Vorgänge, die für den externen Schlüssel aktiviert sind, der dem KMS-Schlüssel zugeordnet ist. Dieses Feld wird nur für KMS-Schlüssel in einem externen Schlüsselspeicher angezeigt.

Schlüsselrichtlinie

Wo: Registerkarte für Schlüsselrichtlinie

Steuert den Zugriff auf den KMS-Schlüssel zusammen mit IAM-Richtlinien und Erteilungen. Jeder KMS-Schlüssel besitzt eine Schlüsselrichtlinie. Es ist das einzige obligatorische Berechtigungselement. Um die Schlüsselrichtlinie eines kundenverwalteten KMS-Schlüssels zu ändern, wählen Sie auf der Registerkarte Key policy (Schlüsselrichtlinie) die Option Edit (Bearbeiten). Details hierzu finden Sie unter the section called “Schlüsselrichtlinien”.

Schlüsselrotation

Wo: Registerkarte für Schlüsseldrehung

Aktiviert und deaktiviert die automatischen Drehung der Schlüsselinformationen in einem kundenverwalteten KMS-Schlüssel. Um den Schlüsseldrehungs-Status eines kundenverwalteten KMS-Schlüssels zu ändern, verwenden Sie das Kontrollkästchen auf der Registerkarte Key rotation (Schlüsseldrehung).

Sie können die Drehung von Schlüsselmaterial in einem Von AWS verwalteter Schlüssel nicht aktivieren oder deaktivieren. Von AWS verwaltete Schlüssel werden automatisch jedes Jahr gedreht.

Schlüsselspezifikation

Wo: Registerkarte für kryptografische Konfiguration

Die Art des Schlüsselmaterials im KMS-Schlüssel. AWS KMS unterstützt KMS-Schlüssel mit symmetrischer Verschöüsselung (SYMMMETRIC_DEFAULT), HMAC-KMS-Schlüssel unterschiedlicher Länge, KMS-Schlüssel für RSA-Schlüssel unterschiedlicher Länge und Elliptic-Curve-Schlüssel mit unterschiedlichen Kurven. Details hierzu finden Sie unter Schlüsselspezifikation.

Schlüsseltyp

Wo: Registerkarte für kryptografische Konfiguration

Gibt an, ob der KMS-Schlüssel symmetrisch oder asymmetrisch ist.

Schlüsselnutzung

Wo: Registerkarte für kryptografische Konfiguration

Gibt an, ob ein KMS-Schlüssel für Encrypt and decrypt (Verschlüsseln und Entschlüsseln), Sign and verify (Signieren und Überprüfen) und Generate and verify MAC (MAC generieren und überprüfen) verwendet werden kann. Details hierzu finden Sie unter Schlüsselnutzung.

Ursprung

Wo: Registerkarte für kryptografische Konfiguration

Die Quelle des Schlüsselmaterials für den KMS-Schlüssel. Gültige Werte für sind:

- AWS KMS für Schlüsselmaterial, das AWS KMS generiert
- AWS CloudHSM für KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher
- Extern für importiertes Schlüsselmaterial (BYOK)
- Externer Schlüsselspeicher für KMS-Schlüssel in einem externen Schlüsselspeicher

MAC-Algorithmen

Wo: Registerkarte für kryptografische Konfiguration

Listet die MAC-Algorithmen auf, die mit einem HMAC-KMS-Schlüssel in AWS KMS verwendet werden können. Dieses Feld wird nur angezeigt, wenn Key spec (Schlüsselspezifikation) eine HMAC-Schlüsselspezifikation (HMAC_*) ist. Weitere Informationen zu den von AWS KMS

unterstützten MAC-Algorithmen finden Sie unter Schlüsselspezifikationen für HMAC-KMS-Schlüssel.

Primärschlüssel

Wo: Registerkarte für Regionalität

Gibt an, dass dieser KMS-Schlüssel ein multiregionaler Primärschlüssel ist. Autorisierte Benutzer können diesen Abschnitt zum Ändern des Primärschlüssels in einen anderen verwandten multiregionalen Schlüssel verwenden. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel ein multiregionaler Primärschlüssel ist.

Öffentlicher Schlüssel

Wo: Registerkarte für öffentliche Schlüssel

Zeigt den öffentlichen Schlüssel eines asymmetrischen KMS-Schlüssels an. Autorisierte Benutzer können diese Registerkarte verwenden, um den öffentlichen Schlüssel zu kopieren und herunterzuladen.

Regionalität

Wo: Abschnitt über allgemeine Konfiguration und Registerkarten für Regionalität

Gibt an, ob ein KMS-Schlüssel ein einzelregionaler Schlüssel, ein multiregionaler Primärschlüssel oder ein multiregionaler Replikatschlüssel ist. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel ein multiregionaler Schlüssel ist.

Verwandte multiregionale Schlüssel

Wo: Registerkarte für Regionalität

Zeigt alle verwandte multiregionale Primär- und Replikatschlüssel an, mit Ausnahme des aktuellen KMS-Schlüssels. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel ein multiregionaler Schlüssel ist.

Im Abschnitt über verwandte multiregionale Schlüssel eines Primärschlüssels können autorisierte Benutzer neue Replikatschlüssel erstellen.

Replikat-Schlüssel

Wo: Registerkarte für Regionalität

Gibt an, dass dieser KMS-Schlüssel ein multiregionaler Replikatschlüssel ist. Dieses Feld wird nur angezeigt, wenn der KMS-Schlüssel ein multiregionaler Replikatschlüssel ist.

Signaturalgorithmen

Wo: Registerkarte für kryptografische Konfiguration

Listet die Signaturalgorithmen auf, die mit dem KMS-Schlüssel in AWS KMS verwendet werden können. Dieses Feld wird nur angezeigt, wenn Key type (Schlüsseltyp) auf Asymmetric (Asymmetrisch) und Key usage (Schlüsselnutzung) auf Sign and verify (Signieren und Überprüfen) eingestellt ist. Weitere Informationen zu den von AWS KMS unterstützten Signaturalgorithmen finden Sie unter RSA-Schlüsselspezifikationen für Signatur und Verifizierung und Elliptic Curve(EC)-Schlüsselspezifikationen.

Status

Wo: Abschnitt über allgemeine Konfiguration

Der Schlüsselstatus des KMS-Schlüssels. Sie können den KMS-Schlüssel nur in kryptografischen Operationen verwenden, wenn der Status Enabled (Aktiviert) lautet. Eine detaillierte Beschreibung eines jeden KMS-Schlüssel-Status und seiner Auswirkungen darauf, welche Operationen Sie für den KMS-Schlüssel ausführen können, finden Sie unter Wichtige Zustände von AWS KMS Schlüsseln.

Tags

Wo: Registerkarte für Tags

Optionale Schlüssel-Wert-Paare, die den KMS-Schlüssel beschreiben. Um die Tags für einen KMS-Schlüssel hinzuzufügen oder zu ändern, wählen Sie auf der Tags-Registerkarte die Option Edit (Bearbeiten).

Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

Anpassen Ihrer KMS-Schlüsseltabellen

Sie können die Tabellen, die auf den Seiten Von AWS verwaltete Schlüssel und Customer managed keys (Vom Kunden verwaltete Schlüssel) in der AWS Management Console angezeigt werden, gemäß Ihren Anforderungen anpassen. Sie können die Tabellenspalten, die Anzahl der AWS KMS keys auf jeder Seite (Seitengröße) und den Textumbruch auswählen. Die von Ihnen gewählte

Konfiguration wird gespeichert, wenn Sie sie bestätigen, und bei jedem Öffnen der Seiten erneut angewendet.

Anpassen Ihrer KMS-Schlüsseltabellen

1. Wählen Sie auf der Seite Von AWS verwaltete Schlüssel oder Customer managed keys (Vom Kunden verwaltete Schlüssel) das Einstellungen-Symbol () oben rechts auf der Seite aus.
2. Wählen Sie auf der Seite Preferences (Einstellungen) die gewünschten Einstellungen aus und wählen Sie dann Confirm (Bestätigen) aus.

Erwägen Sie, mit der Einstellung Page size (Seitengröße) die Anzahl der auf jeder Seite angezeigten KMS-Schlüssel zu erhöhen, insbesondere wenn Sie normalerweise ein Gerät verwenden, das leicht zu scrollen ist.

Die angezeigten Datenspalten können abhängig von der Tabelle, Ihrer Aufgabenrolle und den Typen von KMS-Schlüsseln im Konto und in der Region variieren. Die folgende Tabelle enthält einige vorgeschlagene Konfigurationen. Beschreibungen der Spalten finden Sie unter Anzeigen von KMS-Schlüsseldetails.

Empfohlene Konfigurationen für KMS-Schlüsseltabellen

Sie können die Spalten anpassen, die in der KMS-Schlüsseltabelle angezeigt werden, damit die benötigten KMS-Schlüsseldaten sichtbar werden.

Von AWS verwaltete Schlüssel

Standardmäßig enthält die Von AWS verwalteter Schlüssel-Tabelle die Spalten Aliases (Aliase), Key ID (Schlüssel-ID) und Status. Diese Spalten sind für die meisten Anwendungsfälle gut geeignet.

KMS-Schlüssel mit symmetrischer Verschlüsselung

Wenn Sie nur KMS-Schlüssel mit symmetrischer Verschlüsselung mit Schlüsselmaterial verwenden, das von AWS KMS generiert wurde, sind die Spalten Aliases (Aliase), Key ID (Schlüssel-ID), Status und Creation date (Erstellungsdatum) wahrscheinlich am nützlichsten.

Asymmetrische KMS-Schlüssel

Wenn Sie asymmetrische KMS-Schlüssel verwenden, sollten Sie zusätzlich zu den Spalten Aliases (Aliase), Key ID (Schlüssel-ID) und Status die Spalten Key type (Schlüsseltyp), Key spec (Schlüsselspezifikation) und Key usage (Schlüsselnutzung) hinzufügen. In diesen Spalten wird angezeigt, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, der Typ des Schlüsselmaterials und ob der KMS-Schlüssel für die Verschlüsselung oder Signatur verwendet werden kann.

HMAC-KMS-Schlüssel

Wenn Sie HMAC-KMS-Schlüssel verwenden, sollten Sie zusätzlich zu den Spalten Aliases (Aliase), Key ID (Schlüssel-ID) und Status das Hinzufügen der Spalten Key type (Schlüsseltyp), Key spec (Schlüsselspezifikation) und Key usage (Schlüsselnutzung) erwägen. Diese Spalten zeigen Ihnen, ob ein KMS-Schlüssel ein HMAC-Schlüssel ist. Da Sie KMS-Schlüssel nicht nach Schlüsselspezifikation oder Schlüsselverwendung sortieren können, identifizieren Sie Ihre HMAC-Schlüssel mithilfe von Aliasen und Tags und verwenden dann die Filter-Funktionen der AWS KMS-Konsole, um nach Aliasen oder Tags zu filtern.

Importiertes Schlüsselmaterial

Wenn Sie KMS-Schlüssel mit importiertem Schlüsselmaterial haben, erwägen Sie, die Spalten Origin (Ursprung) und Expiration date (Ablaufdatum) hinzuzufügen. Diese Spalten zeigen Ihnen, ob das Schlüsselmaterial in einem KMS-Schlüssel importiert oder von AWS KMS generiert ist und wann das Schlüsselmaterial abläuft, sofern zutreffend. Das Feld Creation date (Erstellungsdatum) zeigt das Datum an, an dem der KMS-Schlüssel erstellt wurde (ohne Schlüsselmaterial). Es spiegelt keine Eigenschaften des Schlüsselmaterials wider.

Schlüssel in benutzerdefinierten Schlüsselspeichern

Wenn Sie KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern haben, erwägen Sie, die Spalten Origin (Ursprung) und Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) hinzuzufügen. Diese Spalten geben an, dass sich der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher befindet, zeigen den Typ des benutzerdefinierten Schlüsselspeichers an und identifizieren den benutzerdefinierten Schlüsselspeicher.

Multiregionale Schlüssel

Wenn Sie multiregionale Schlüssel haben, erwägen Sie, die Spalte Regionality (Regionalität) hinzuzufügen. Gibt an, ob ein KMS-Schlüssel ein einzelregionaler Schlüssel, ein multiregionaler Primärschlüssel oder ein multiregionaler Replikatschlüssel ist.

Anzeigen von KMS-Schlüssel mit der API

Sie können die AWS Key Management Service (AWS KMS)-API zum Anzeigen Ihrer KMS-Schlüssel verwenden. In diesem Abschnitt werden verschiedene Operationen demonstriert, die Details zu vorhandenen KMS-Schlüssel liefern. Für die Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Themen

ListKeys: Abrufen der ID und des ARN aller KMS-Schlüssel

Die ListKeys Operation gibt die ID und den Amazon-Ressourcennamen (ARN) aller KMS-Schlüssel im Konto und in der Region zurück.

Dieser Aufruf der `ListKeys`-Produktion gibt beispielsweise die ID und den ARN jedes KMS-Schlüssels in diesem fiktiven Konto zurück. Beispiele in verschiedenen Programmiersprachen finden Sie unter Abruf von Schlüssel-IDs und Schlüssel-ARNs von KMS-Schlüsseln.

```
$ aws kms list-keys

{
  "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"
    }
}
```

DescribeKey: Abrufen detaillierter Informationen zu einem KMS-Schlüssel

Die DescribeKey Operation gibt Details zum angegebenen KMS-Schlüssel zurück. Verwenden Sie zum Identifizieren eines KMS-Schlüssels seine Schlüssel-ID, den Schlüssel-ARN, den Aliasnamen oder den Alias-ARN.

Im Gegensatz zu der -ListKeysOperation, die nur KMS-Schlüssel im Konto und in der Region des Aufrufers anzeigt, können autorisierte Benutzer die -`DescribeKey`Operation verwenden, um Details zu KMS-Schlüsseln in anderen Konten abzurufen.

> **Note**
>
> Die `DescribeKey`-Antwort enthält sowohl `KeySpec` und `CustomerMasterKeySpec`-Elemente mit den gleichen Werten. Das `CustomerMasterKeySpec`-Element ist veraltet.

Dieser Aufruf an `DescribeKey` gibt beispielsweise Informationen zu einem KMS-Schlüssel mit symmetrischer Verschlüsselung zurück. Die Felder in der Antwort variieren je nach AWS KMS key-Spezifikation, Schlüsselstatus und Ursprung des Schlüsselmaterials. Beispiele in verschiedenen Programmiersprachen finden Sie unter Anzeigen eines AWS KMS key.

```
$ 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"
        ]
    }
}
```

In diesem Beispiel wird die `DescribeKey`-Produktion für einen asymmetrischen KMS-Schlüssel aufgerufen, der für Signatur und Verifizierung verwendet wird. Die Antwort enthält die Signaturalgorithmen, die AWS KMS für diesen KMS-Schlüssel unterstützt.

```
$ 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"
        ]
    }
}
```

GetKeyPolicy: Abrufen der Schlüsselrichtlinie, die einem KMS-Schlüssel zugeordnet ist

Der GetKeyPolicy Vorgang ruft die Schlüsselrichtlinie ab, die dem KMS-Schlüssel angefügt ist. Verwenden Sie zum Identifizieren des KMS-Schlüssels seine Schlüssel-ID oder den Schlüssel-ARN.

Sie müssen außerdem den Richtliniennamen (immer `default`) angeben. (Wenn Ihre Ausgabe schwer lesbar ist, fügen Sie `--output text`-Option zu Ihrem Befehl hinzu.) `GetKeyPolicy` funktioniert nur bei KMS-Schlüsseln im Konto und in der Region des Anrufers.

Beispiele in verschiedenen Programmiersprachen finden Sie unter Abrufen einer Schlüsselrichtlinie.

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

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

ListAliases: Abrufen von Aliasnamen und ARNs für KMS-Schlüssel

Der ListAliases Vorgang gibt Aliase im Konto und in der Region zurück. Der `TargetKeyId`-Wert in der Antwort enthält ggf. die Schlüssel-ID des KMS-Schlüssels, auf den sich der Alias bezieht.

Standardmäßig gibt der Befehl ListAliases alle Aliase innerhalb des Kontos und der Region zurück. Darunter fallen auch Aliase, die Sie erstellt und Ihren vom Kunden verwalteten Schlüsseln zugeordnet haben, sowie Aliase, die AWS erstellt und demVon AWS verwalteter Schlüssel in Ihrem Konto zugeordnet hat. AWS-Aliase sind daran zu erkennen, dass ihre Namen das Format `aws/<service-name>` aufweisen, z. B. `aws/dynamodb`.

Die Antwort kann auch Aliase ohne dem `TargetKeyId`-Feld enthalten, z. B. der Alias `aws/redshift` in diesem Beispiel. Dies sind vordefinierte Aliasse, die von AWS erstellt, aber noch keinem KMS-Schlüssel zugeordnet wurden.

Beispiele in verschiedenen Programmiersprachen finden Sie unter Auflisten von Aliasen.

```
$ 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/financeKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "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/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
        },
```

```
        {
            "AliasName": "alias/aws/redshift",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/redshift"
        },
    ]
}
```

Um die Aliase zu erhalten, die auf einen bestimmten KMS-Schlüssel verweisen, verwenden Sie den `KeyId`-Parameter. Der Parameterwert kann die Schlüssel-ID oder der Schlüssel-ARN sein. Sie können keinen Aliasnamen oder Alias-ARN angeben.

Mit dem Befehl im folgenden Beispiel werden die Aliase abgerufen, die auf einen kundenverwalteten Schlüssel verweisen. Sie können mit einem entsprechenden Befehl jedoch auch die Aliase finden, die sich auf einen Von AWS verwaltete Schlüssel beziehen.

```
$ aws kms list-aliases --key-id arn:aws:kms:us-
west-2:111122223333:key/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": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "AliasName": "alias/financeKey",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
    ]
}
```

Um nur die Aliase für Von AWS verwaltete Schlüssel abzurufen, filtern Sie mithilfe der Funktionen Ihrer Programmiersprache die Antwort.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

ListResourceTags: Abrufen der Tags für KMS-Schlüssel

Die ListResourceTags Operation gibt die Tags für den angegebenen KMS-Schlüssel zurück. Die API gibt Tags für einen KMS-Schlüssel zurück. Sie können den Befehl jedoch in einer Schleife ausführen, um Tags für alle KMS-Schlüssel im Konto und in der Region oder für eine Reihe von ausgewählten KMS-Schlüsseln abzurufen. Diese API gibt jeweils eine Seite zurück. Wenn Sie also zahlreiche Tags auf zahlreichen KMS-Schlüsseln haben, müssen Sie möglicherweise die Paginierung in Ihrer Programmiersprache verwenden, um alle gewünschten Tags zu erhalten.

Die `ListResourceTags`-Operation gibt Tags für alle KMS-Schlüssel zurück, Von AWS verwalteter Schlüssel sind jedoch nicht getaggt. Es funktioniert nur mit KMS-Schlüsseln im Konto und in der Region des Anrufers.

Um die Tags für einen KMS-Schlüssel zu finden, verwenden Sie die `ListResourceTags`-Produktion. Der Parameter `KeyId` muss angegeben werden. Sie akzeptiert eine Schlüssel-ID oder Schlüssel-ARN. Ersetzen Sie vor Ausführung dieses Beispiels den Beispiel-Schlüssel-ARN durch einen gültigen.

```
$  aws kms list-resource-tags --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
      {
    "Tags": [
        {
            "TagKey": "Department",
            "TagValue": "IT"
        },
        {
            "TagKey": "Purpose",
            "TagValue": "Test"
        }
    ],
    "Truncated": false
}
```

Sie können die `ListResourceTags`-Produktion verwenden, um alle KMS-Schlüssel im Konto und in der Region mit einem bestimmten Tag, Tag-Schlüssel oder Tag-Wert abzurufen. Um dies zu tun, verwenden Sie die Filterfunktionen Ihrer Programmiersprache.

Das folgende Bash-Skript verwendet beispielsweise die `ListResourceTags` Operationen ListKeys und , um alle KMS-Schlüssel im Konto und in der Region mit einem `Project` Tag-Schlüssel

abzurufen. Beide Operationen erhalten nur die erste Seite der Ergebnisse. Wenn Sie zahlreiche KMS-Schlüssel oder zahlreiche Tags haben, verwenden Sie die Paginierungsfunktionen Ihrer Sprache, um das gesamte Ergebnis aus jeder Produktion zu erhalten. Ersetzen Sie vor Ausführung dieses Beispiels den Beispiel-Schlüssel-IDs durch gültige.

```
TARGET_TAG_KEY='Project'

for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text); do
  key_tags=$(aws kms list-resource-tags --key-id "$key" --query "Tags[?TagKey==\`
$TARGET_TAG_KEY\`]")
  if [ "$key_tags" != "[]" ]; then
    echo "Key: $key"
    echo "$key_tags"
  fi
done
```

Die Ausgabe wird wie die folgende Beispielausgabe formatiert.

```
Key: 0987dcba-09fe-87dc-65ba-ab0987654321
[
    {
        "TagKey": "Project",
        "TagValue": "Gamma"
    }
]
Key: 1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
Key: 0987ab65-43cd-21ef-09ab-87654321cdef
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
```

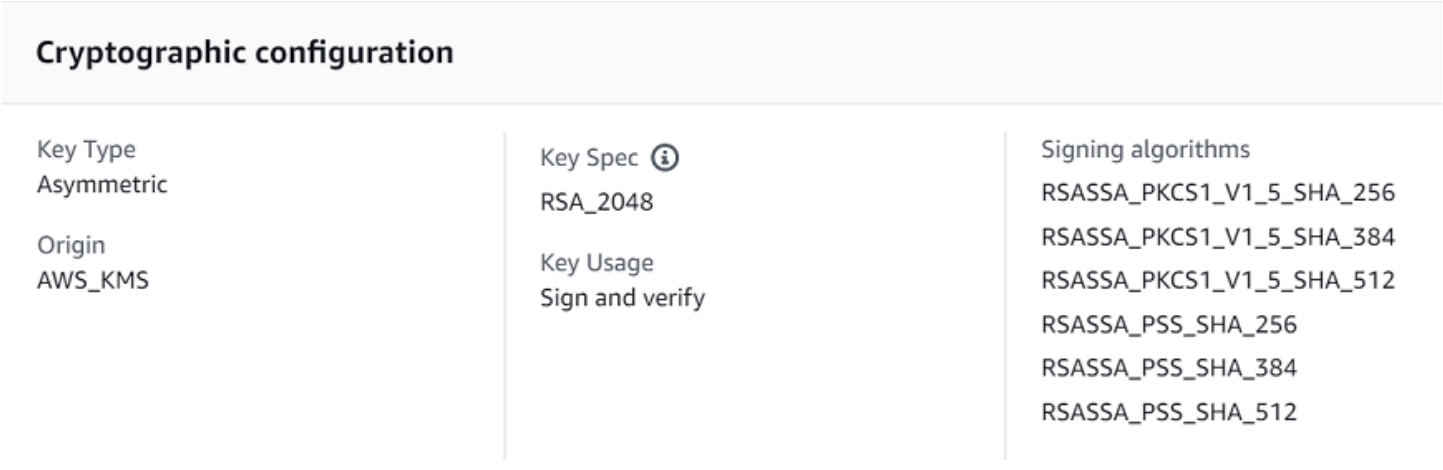
Anzeigen der kryptografischen Konfiguration von KMS-Schlüsseln

Nachdem Sie Ihren KMS-Schlüssel erstellt haben, können Sie seine kryptografische Konfiguration anzeigen. Die Konfiguration einmal erstellter KMS-Schlüssel kann nicht mehr geändert werden. Wenn Sie eine andere Konfiguration bevorzugen, löschen Sie den KMS-Schlüssel und erstellen Sie ihn erneut.

Sie können in der AWS KMS-Konsole oder mithilfe der AWS KMS-API die kryptografische Konfiguration Ihrer KMS-Schlüssel finden und Schlüsselspezifikation, Schlüsselnutzung und unterstützte Verschlüsselungs- oder Signaturalgorithmen einschließen. Details hierzu finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

In der AWS KMS-Konsole enthält die Detailseite eines jeden KMS-Schlüssels die Registerkarte Cryptographic configuration (kryptografische Konfiguration), auf der kryptografische Details zu Ihren KMS-Schlüsseln angezeigt werden. Die folgende Abbildung zeigt beispielsweise die Registerkarte Cryptographic configuration (kryptografische Konfiguration) für einen RSA-KMS-Schlüssel, der für Signierung und Verifizierung verwendet wird.

Die Registerkarte Cryptographic configuration (Kryptografische Konfiguration) für einige spezielle KMS-Schlüssel enthält zusätzliche spezielle Abschnitte. Beispielsweise enthält die Registerkarte Cryptographic configuration (Kryptografische Konfiguration) für einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher einen Abschnitt Custom key stores (Benutzerdefinierte Schlüsselspeicher). Die Registerkarte Cryptographic configuration (Kryptografische Konfiguration) für einen KMS-Schlüssel in einem externen Schlüsselspeicher enthält einen Abschnitt External key (Externer Schlüssel).



Verwenden Sie in der AWS KMS API die -DescribeKeyOperation. Die `KeyMetadata`-Struktur in der Antwort umfasst die kryptografische Konfiguration des KMS-Schlüssels. Beispielsweise gibt

`DescribeKey` die folgende Antwort für einen RSA-KMS-Schlüssel zurück, der für Signierung und Verifizierung verwendet wird.

```
{
  "KeyMetadata": {

    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": 1571767572.317,
    "CustomerMasterKeySpec": "RSA_2048",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "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"
    ]
  }
}
```

Finden der Schlüssel-ID und des Schlüssel-ARN

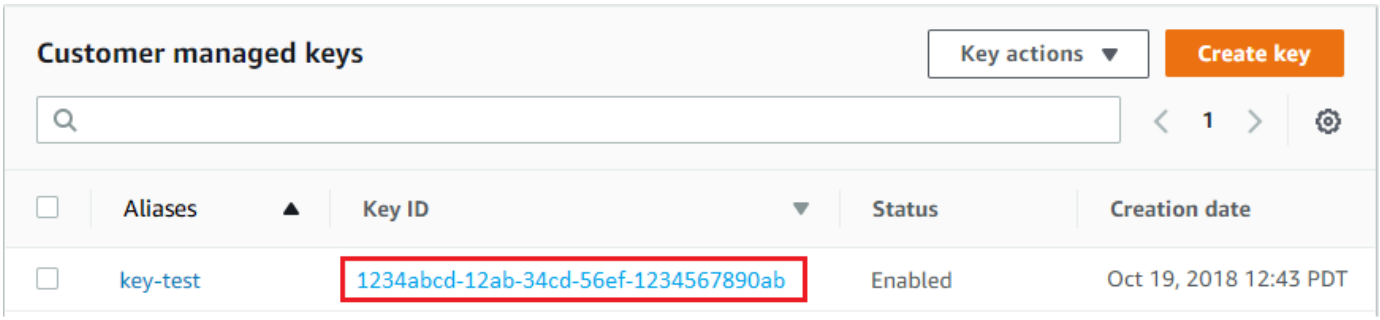
Um einen AWS KMS key-KMS-Schlüssel zu identifizieren, können Sie die Schlüssel-ID oder den Amazon-Ressourcennamen (Schlüssel-ARN) verwenden. Bei kryptografischen Produktionen können Sie auch den Aliasnamen oder den Alias-ARNverwenden.

Ausführliche Informationen zu den von AWS KMS unterstützten KMS-Schlüsselbezeichnern finden Sie unter Schlüsselkennungen (KeyId). Weitere Informationen zur Suche nach Aliasnamen und Alias-ARN finden Sie unter Suchen des Aliasnamens und des Alias-ARN.

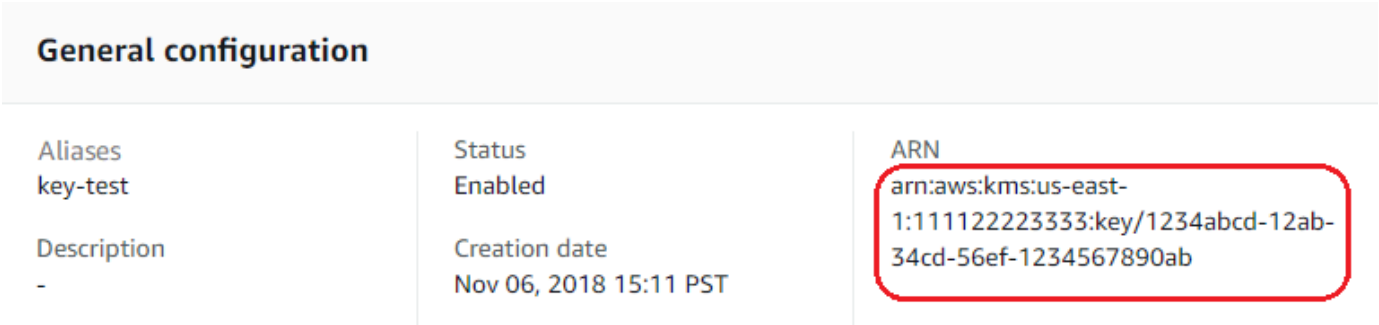
Die Schlüssel-ID und den Schlüssel-ARN finden (Konsole)

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus.
4. Um die Schlüssel-ID für einen KMS-Schlüssel zu finden, suchen Sie nach der Zeile, die mit dem KMS-Schlüssel-Alias beginnt.

   Die Spalte Key ID (Schlüssel-ID) wird standardmäßig in den Tabellen angezeigt. Wenn die Schlüssel-ID-Spalte nicht in der Tabelle angezeigt wird, verwenden Sie das unter the section called “Anpassen Ihrer KMS-Schlüsseltabellen” beschriebene Verfahren, um sie wiederherzustellen. Sie können die Schlüssel-ID eines KMS-Schlüssels auch auf der Detailseite anzeigen.



5. Um den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels zu suchen, wählen Sie die Schlüssel-ID oder den Alias aus. Der Schlüssel-ARN wird im Bereich General Configuration (Allgemeine Konfiguration) angezeigt.



Die Schlüssel-ID und den Schlüssel-ARN finden (AWS KMS-API)

Um die Schlüssel-ID und den Schlüssel-ARN eines zu findenAWS KMS key, verwenden Sie die -ListKeysOperation. Beispiele in verschiedenen Programmiersprachen finden Sie unter Abruf von Schlüssel-IDs und ARNs und Abruf von Schlüssel-IDs und ARNs.

Die `ListKeys` Antwort enthält die Schlüssel-ID und den Schlüssel-ARN für jeden KMS-Schlüssel im Konto und in der Region.

```
$ 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"
        }
    ]
}
```

Suchen des Aliasnamens und des Alias-ARN

Ein Alias ist ein Anzeigename für einen AWS KMS AWS KMS keys (KMS-Schlüssel). Sie finden den Aliasnamen und Alias-ARN in der AWS KMS-Konsole oderAWS KMS-API.

Ausführliche Informationen zu den von AWS KMS unterstützten KMS-Schlüsselbezeichnern finden Sie unter Schlüsselkennungen (KeyId). Hilfestellung beim Suchen der Schlüssel-ID und des Schlüssel-ARN eines KMS-Schlüssels finden Sie unter Finden der Schlüssel-ID und des Schlüssel-ARN.

Themen

- Den Aliasnamen und den Alias-ARN finden (Konsole)
- Den Aliasnamen und den Alias-ARN finden (AWS KMS-API)

Den Aliasnamen und den Alias-ARN finden (Konsole)

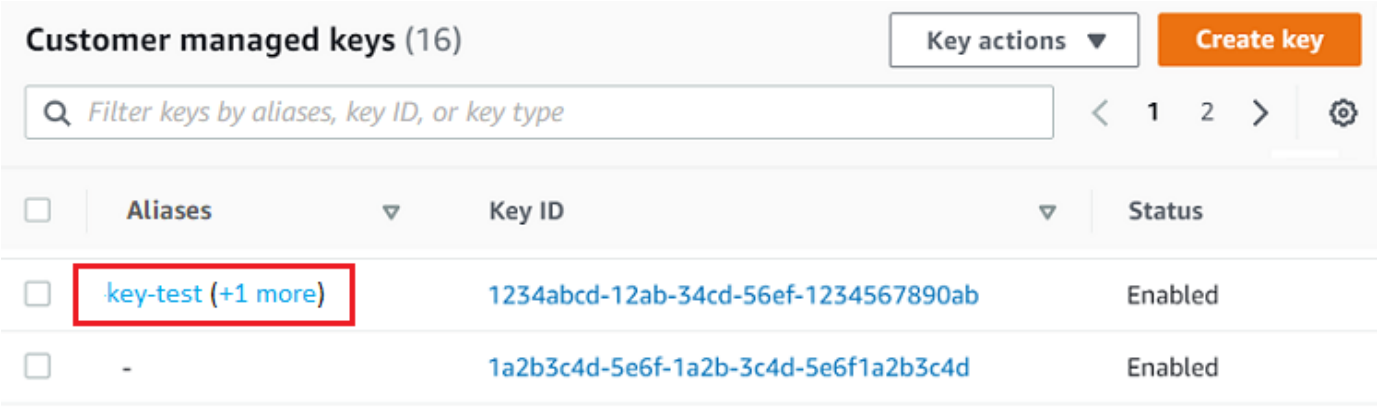
Die AWS KMS-Konsole zeigt die Aliase an, die dem KMS-Schlüssel zugeordnet sind.

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus.
4. Die Spalte Aliases (Aliase) zeigt den Alias für jeden KMS-Schlüssel an. Wenn ein KMS-Schlüssel keinen Alias hat, wird ein Bindestrich (-) in der Spalte Aliases (Aliase) angezeigt.

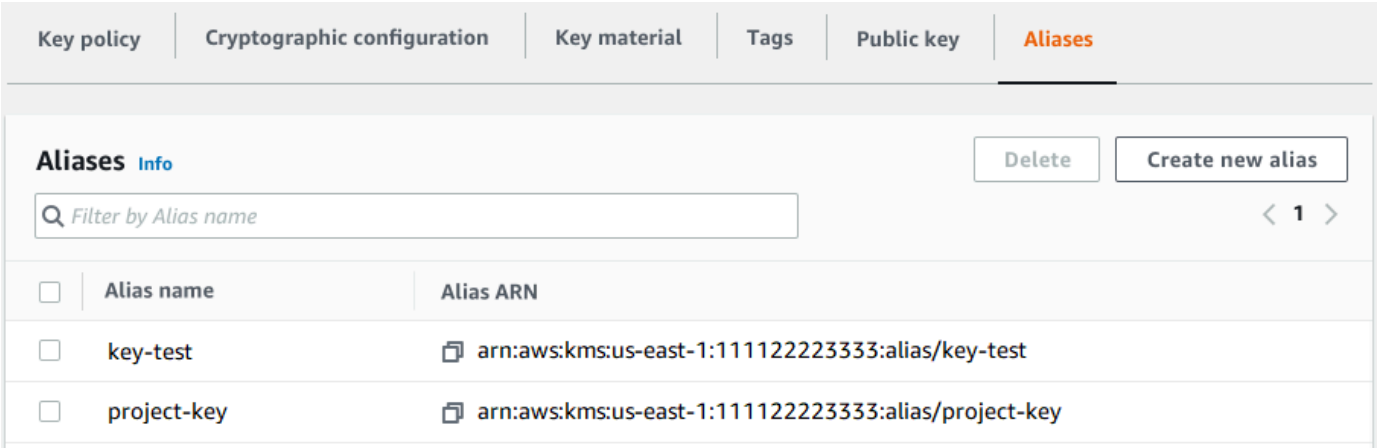
   Wenn ein KMS-Schlüssel über mehrere Aliase verfügt, enthält die Aliase-Spalte auch eine Aliasübersicht, z. B. (+n mehr). Der folgende KMS-Schlüssel hat beispielsweise zwei Aliase, von denen einer ist `key-test`.

   Um den Aliasnamen und den Alias-ARN aller Aliase für den KMS-Schlüssel zu finden, verwenden Sie die Aliase-Registerkarte.

   - Um direkt zur Registerkarte Aliases (Aliase) zu gelangen, wählen Sie in der Spalte Aliases (Aliase) die Aliasübersicht aus (+n mehr). Eine Aliasübersicht wird nur angezeigt, wenn der KMS-Schlüssel mehr als einen Alias hat.
   - Oder wählen Sie den Alias oder die Schlüssel-ID des KMS-Schlüssels aus (wodurch die Detailseite für den KMS-Schlüssel geöffnet wird) und wählen Sie dann die Aliase-Registerkarte. Die Registerkarten werden unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.



5. Die Aliase-Registerkarte zeigt den Aliasnamen und Alias-ARN aller Aliase für einen KMS-Schlüssel an. Auf dieser Registerkarte können Sie auch Aliase für den KMS-Schlüssel erstellen und löschen.



Den Aliasnamen und den Alias-ARN finden (AWS KMS-API)

Verwenden Sie die Operation AWS KMS key, um den Aliasnamen und den Alias-ARN eines zu findenListAliases. Beispiele in verschiedenen Programmiersprachen finden Sie unter Auflisten von Aliasen und Abruf von Aliasnamen und ARNs.

Standardmäßig enthält die Antwort den Aliasnamen und den Alias-ARN für jeden Alias im Konto und in der Region. Um nur die Aliase für einen bestimmten KMS-Schlüssel abzurufen, verwenden Sie den `KeyId`-Parameter.

Der folgende Befehl ruft beispielsweise nur die Aliase für einen Beispiel-KMS-Schlüssel mit der Schlüssel-ID `1234abcd-12ab-34cd-56ef-1234567890ab` ab.

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/key-test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/key-test",
            "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
        }
    ]
}
```

Bearbeiten von Schlüsseln

Sie können die folgenden Eigenschaften Ihrer kundenverwalteter Schlüssel in der AWS KMS-Konsole und durch Verwendung des AWS KMS-API ändern.

Sie können keine Eigenschaften von Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel bearbeiten. Diese Schlüssel werden von den AWS-Services verwaltet, die sie erstellt haben.

Beschreibung

Sie können die Beschreibung Ihres vom Kunden verwalteten Schlüssels auf der Detailseite für den KMS-Schlüssel oder mithilfe der -UpdateKeyDescriptionOperation ändern.

Wählen Sie zum Bearbeiten der Schlüsselbeschreibung in der Konsole in der rechten oberen Ecke der Detailseite für den KMS-Schlüssel Edit (Bearbeiten)aus.

Schlüsselrichtlinie

Sie können die Schlüsselrichtlinie auf der Registerkarte Schlüsselrichtlinie der Detailseite für den vom Kunden verwalteten Schlüssel oder mithilfe der -PutKeyPolicyOperation ändern.

Details hierzu finden Sie unter Ändern einer Schlüsselrichtlinie.

Tags

Sie können Tags auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) der AWS KMS-Konsole oder auf der Registerkarte Tags der Detailseite für den kundenverwalteten Schlüssel anlegen und löschen. Oder Sie können die UntagResource Operationen TagResource und verwenden.

Details hierzu finden Sie unter Tagging von Schlüsseln.

Aktivieren und Deaktivieren

Sie können KMS-Schlüssel auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) der AWS KMS-Konsole oder auf der Detailseite für den kundenverwalteten Schlüssel aktivieren oder deaktivieren. Oder Sie können die DisableKey Operationen EnableKey und verwenden.

Details hierzu finden Sie unter Aktivieren und Deaktivieren von Schlüsseln.

Automatische Schlüsselrotation

Sie können die automatische Schlüsseldrehung auf der Registerkarte Schlüsseldrehung der Detailseite für den vom Kunden verwalteten Schlüssel oder mithilfe der - EnableKeyRotation und -DisableKeyRotationOperationen aktivieren und deaktivieren.

Details hierzu finden Sie unter Rotierend AWS KMS keys.

Weitere Informationen finden Sie auch unter

Aktualisieren von Aliassen

Tagging von Schlüsseln

In AWS KMS können Sie Tags zu einem kundenverwaltete Schlüssel hinzufügen wenn Sie den KMS-Schlüssel erstellen, und vorhandene KMS-Schlüssel markieren oder entmarkieren, es sei denn, dessen Löschung steht aus. Sie können keine Aliasse, benutzerdefinierten Schlüsselspeicher, Von AWS verwaltete Schlüssel, AWS-eigene Schlüssel oder KMS-Schlüssel in anderen AWS-Konten markieren. Tags sind optional, aber sie können sehr nützlich sein.

Weitere Informationen finden Sie unter Erstellen von Schlüsseln und Bearbeiten von Schlüsseln. Allgemeine Informationen zu Tags, einschließlich bewährter Methoden, Markierungs-Strategien und das Format und die Syntax von Tags, finden Sie unter Markieren von AWS-Ressourcen in Allgemeine Amazon Web Services-Referenz.

Themen

Informationen zu Tags in AWS KMS

Ein Tag ist ein optionales Metadaten-Etikett, das von Ihnen oder von AWS einer AWS-Ressource zugewiesen wird. Jedes Tag besteht aus einem Tag-Schlüssel und einem Tag-Wert, wobei beide Zeichenfolgen zwischen Groß-/Kleinschreibung unterscheiden. Der Wert kann auch eine leere (Null) Zeichenfolge sein. Jedes Tag auf einer Ressource muss über einen anderen Tag-Schlüssel verfügen, Sie können jedoch dasselbe Tag mehreren AWS-Ressourcen hinzufügen. Eine Ressource kann bis zu 50 Tags besitzen, die von Benutzern erstellt wurden.

Geben Sie keine vertraulichen oder sensiblen Informationen in den Tag-Schlüssel oder Tag-Wert ein. Tags sind für viele AWS-Services zugänglich, einschließlich der Abrechnung.

In AWS KMS können Sie Tags zu einem kundenverwaltete Schlüssel hinzufügen wenn Sie den KMS-Schlüssel erstellen, und vorhandene KMS-Schlüssel markieren oder entmarkieren, es sei denn, dessen Löschung steht aus. Sie können keine Aliasse, benutzerdefinierten Schlüsselspeicher, Von AWS verwaltete Schlüssel, AWS-eigene Schlüssel oder KMS-Schlüssel in anderen AWS-Konten markieren. Tags sind optional, aber sie können sehr nützlich sein.

Beispielsweise können Sie ein `"Project"="Alpha"`-Tag allen KMS-Schlüsseln und Amazon-S3-Buckets hinzufügen, die Sie für das Alpha-Projekt verwenden.

```
TagKey   = "Project"
TagValue = "Alpha"
```

Allgemeine Informationen zu Tags, einschließlich dem Format und der Syntax, finden Sie unter Markieren von AWS-Ressourcen in Allgemeine Amazon Web Services-Referenz.

Tags sind für folgende Aktivitäten nützlich:

- Identify and organize your AWS resources. Viele AWS-Services unterstützen das Markieren mit Tags (kurz: Tagging). So können Ressourcen aus verschiedenen Services dasselbe Tag zuweisen,

um anzugeben, dass die Ressourcen verbunden sind. Beispielsweise können Sie das gleiche Tag einem KMS-Schlüssel und einem Amazon Elastic Block Store (Amazon EBS)-Volume oder AWS Secrets Manager-Geheimnis zuweisen. Sie können Tags auch zur Identifizierung von KMS-Schlüsseln für die Automatisierung verwenden.

- Überwachen von AWS-Kosten. Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Sie können dieses Feature verwenden, um AWS KMS-Kosten für ein Projekt, eine Anwendung oder eine Kostenstelle zu verfolgen.

  Weitere Informationen zur Verwendung von Tags für die Kostenzuordnung finden Sie unter Verwenden von Kostenzuordnungs-Tags im AWS Billing-Benutzerhandbuch. Weitere Informationen über die Regeln, die für die Tag-Schlüssel und Tag-Werte gelten, finden Sie unter Beschränkungen benutzerdefinierter Tags im AWS Billing-Benutzerhandbuch.
- Kontrollieren Sie den Zugriff auf Ihre AWS-Ressourcen. Das Zulassen und Verweigern des Zugriffs auf KMS-Schlüssel basierend auf ihren Tags ist Teil der AWS KMS-Unterstützung für attributbasierte Zugriffssteuerung (ABAC). Weitere Hinweise zum Steuern des Zugriffs auf AWS KMS keys anhand von ihren Tags finden Sie unter Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel. Weitere Informationen zur Verwendung von Tags, um den Zugriff auf Ihre AWS-Ressourcen zu steuern, finden Sie unter Steuerung des Zugriffs auf AWS-Ressourcen mit Ressourcen-Tags im IAM-Benutzerhandbuch.

AWS KMS schreibt einen Eintrag in Ihr AWS CloudTrail ProtokollTagResource, wenn Sie die ListResourceTags Operationen UntagResource, oder verwenden.

Verwalten von KMS-Schlüssel-Tags in der Konsole

Sie können Tags zu einem KMS-Schlüssel hinzufügen, wenn Sie in der AWS KMS-Konsole den KMS-Schlüssel erstellen. Sie können auch die Tags-Registerkarte in der Konsole verwenden, um Tags für kundenverwaltete Schlüssel hinzuzufügen, zu bearbeiten und zu löschen. Um Tags für einen KMS-Schlüssel hinzuzufügen, zu bearbeiten, anzuzeigen und zu löschen, müssen Sie über die erforderlichen Berechtigungen verfügen. Details hierzu finden Sie unter Steuern des Zugriffs auf Tags.

Hinzufügen von Tags beim Erstellen eines KMS-Schlüssels

Um Tags beim Erstellen eines KMS-Schlüssels in der Konsole hinzuzufügen, müssen Sie die `kms:TagResource`-Berechtigung in einer IAM-Richtlinie haben, zusätzlich zu den Berechtigungen,

die zum Erstellen von KMS-Schlüsseln und zum Anzeigen von KMS-Schlüsseln in der Konsole erforderlich sind. Die Berechtigung muss mindestens alle KMS-Schlüssel im Konto und in der Region abdecken.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. (Die Tags von Von AWS verwalteter Schlüssel können Sie nicht verwalten.)
4. Wählen Sie den Schlüsseltyp aus, und wählen Sie dann Next (Weiter).
5. Geben Sie einen Alias und eine optionale Beschreibung ein.
6. Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wenn Sie zusätzliche Tags hinzufügen möchten, wählen Sie Add new tag (neues Tag hinzufügen) aus. Zum Entfernen eines Tags wählen Sie Remove (Entfernen). Wählen Sie nach dem Markieren des neuen KMS-Schlüssels Next (Weiter).
7. Schließen Sie das Erstellen des KMS-Schlüssels ab.

Anzeigen und Verwalten von Tags auf vorhandenen KMS-Schlüsseln

Um Tags in der Konsole hinzuzufügen, anzuzeigen, zu bearbeiten und zu löschen, benötigen Sie Markierungs-Berechtigung auf dem KMS-Schlüssel. Sie können diese Berechtigung von der Schlüsselrichtlinie für den KMS-Schlüssel oder, wenn die Schlüsselrichtlinie dies zulässt, von einer IAM-Richtlinie abrufen, die den KMS-Schlüssel enthält. Sie benötigen diese Berechtigungen zusätzlich zu den Berechtigungen zum Anzeigen von KMS-Schlüsseln in der Konsole.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. (Die Tags von einem Von AWS verwalteter Schlüssel können Sie nicht verwalten.)
4. Sie können den Tabellenfilter verwenden, um nur KMS-Schlüssel mit bestimmten Tags anzuzeigen. Details hierzu finden Sie unter Sortieren und Filtern Ihrer KMS-Schlüssel.
5. Wählen Sie das Kontrollkästchen neben dem Alias für den KMS-Schlüssel.

6. Wählen Sie Schlüsselaktionen, Hinzufügen und Bearbeiten von Tags.
7. Wählen Sie auf der Detailseite für den KMS-Schlüssel die Tags-Registerkarte.

   - Um Ihr erstes Tag zu erstellen, wählen Sie Create tag (Tag erstellen) aus, geben Sie einen Tag-Schlüssel (erforderlich) und einen Tag-Wert (optional) ein und wählen Sie dann Save (Speichern).

     Wenn Sie den Tag-Wert leer lassen, ist der tatsächliche Tag-Wert eine Null oder eine leere Zeichenfolge.
   - Um ein weiteres Tag hinzuzufügen, wählen Sie Edit (Bearbeiten), wählen Sie dann Add tag (Tag hinzufügen), geben Sie einen Tag-Namen und einen Tag-Wert ein und wählen Sie dann Save (Speichern).
   - Um den Namen oder den Wert eines Tags zu ändern, wählen Sie Edit (Bearbeiten) geben einen Tagnamen und einen Tagwert ein und wählen dann Save (Speichern).
   - Um ein Tag zu löschen, wählen Sie zunächst Edit (Bearbeiten). Wählen Sie dann in der Zeile des Tags die Option Remove (Entfernen) und dann Save (Speichern).
8. Klicken Sie auf Save changes (Änderungen speichern), um die Änderungen zu speichern.

Verwalten von KMS-Schlüsseltags mit API-Operationen

Sie können die AWS Key Management Service (AWS KMS)-API zum Hinzufügen, Löschen und Auflisten von Tags für von Ihnen verwaltete KMS-Schlüssels verwenden. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen. Sie können Von AWS verwaltete Schlüssel nicht markieren.

Um Tags für einen KMS-Schlüssel hinzuzufügen, zu bearbeiten, anzuzeigen und zu löschen, müssen Sie über die erforderlichen Berechtigungen verfügen. Details hierzu finden Sie unter Steuern des Zugriffs auf Tags.

Themen

- CreateKey: Hinzufügen von Tags zu einem neuen KMS-Schlüssel
- TagResource: Hinzufügen oder Ändern von Tags für einen KMS-Schlüssel
- ListResourceTags: Abrufen der Tags für einen KMS-Schlüssel
- UntagResource: Löschen von Tags aus einem KMS-Schlüssel

CreateKey: Hinzufügen von Tags zu einem neuen KMS-Schlüssel

Sie können Tags hinzufügen, wenn Sie einen kundenverwalteten Schlüssel erstellen. Verwenden Sie den -`Tags`Parameter der -CreateKeyOperation, um die Tags anzugeben.

Um Tags beim Erstellen eines KMS-Schlüssels hinzuzufügen, muss der Aufrufer die `kms:TagResource`-Berechtigung in einer IAM-Richtlinie haben. Die Berechtigung muss mindestens alle KMS-Schlüssel im Konto und in der Region abdecken. Details hierzu finden Sie unter Steuern des Zugriffs auf Tags.

Der Wert des `Tags`-Parameters von `CreateKey` ist eine Sammlung von Tag-Schlüssel- und Tag-Wert-Paaren, unter Beachtung der Groß-/Kleinschreibung. Jedes Tag für einen KMS-Schlüssel muss über einen anderen Tag-Namen verfügen. Der Tag-Wert kann auch eine leere (Null) Zeichenfolge sein.

Mit dem folgenden AWS CLI-Befehl wird beispielsweise ein KMS-Schlüssel mit symmetrischer Verschlüsselung mit einem `Project:Alpha`-Tag erstellt. Wenn Sie mehr als ein Schlüssel-Wert-Paar angeben, verwenden Sie ein Leerzeichen, um die einzelnen Paare zu trennen.

```
$ aws kms create-key --tags TagKey=Project,TagValue=Alpha
```

Wenn dieser Befehl erfolgreich ist, gibt er ein `KeyMetadata`-Objekt mit Informationen über den neuen KMS-Schlüssel zurück. Die `KeyMetadata` enthalten jedoch keine Tags. Um die Tags abzurufen, verwenden Sie die -ListResourceTagsOperation.

TagResource: Hinzufügen oder Ändern von Tags für einen KMS-Schlüssel

Die -TagResourceOperation fügt einem KMS-Schlüssel ein oder mehrere Tags hinzu. Sie können diese Operation nicht verwenden, um Tags in einem anderen AWS-Konto hinzuzufügen oder zu bearbeiten.

Geben Sie zum Hinzufügen eines Tags einen neuen Tag-Schlüssel und einen Tag-Wert an. Um ein Tag zu bearbeiten, geben Sie einen vorhandenen Tag-Schlüssel und einen neuen Tag-Wert an. Jedes Tag für einen KMS-Schlüssel muss über einen anderen Tag-Schlüssel verfügen. Der Tag-Wert kann auch eine leere (Null) Zeichenfolge sein.

Mit dem folgenden Befehl werden z. B. die **`Purpose`**- und **`Department`**-Tags zu einem Beispiel-KMS-Schlüssel hinzugefügt.

```
$ aws kms tag-resource \
```

```
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Pretest TagKey=Department,TagValue=Finance
```

Wenn der Befehl erfolgreich war, erfolgt keine Ausgabe. Um die Tags für einen KMS-Schlüssel anzuzeigen, verwenden Sie die -ListResourceTagsOperation.

Sie können auch TagResource verwenden, um den Tag-Wert für ein vorhandenes Tag zu ändern. Um einen Tag-Wert zu ersetzen, geben Sie den gleichen Tag-Schlüssel mit einem unterschiedlichen Wert an.

Beispielsweise ändert dieser Befehl den Wert des `Purpose`-Tag von `Pretest` auf `Test`.

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Test
```

ListResourceTags: Abrufen der Tags für einen KMS-Schlüssel

Der ListResourceTags Vorgang ruft die Tags für einen KMS-Schlüssel ab. Der Parameter `KeyId` muss angegeben werden. Sie können diese Operation nicht verwenden, um Tags für KMS-Schlüssel in einem anderen AWS-Konto anzuzeigen.

Der folgende Befehl ruft die Tags für einen Beispiel-KMS-Schlüssel ab.

```
$ 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"
    }
  ]
```

```
}
```

UntagResource: Löschen von Tags aus einem KMS-Schlüssel

Die -UntagResourceOperation löscht Tags aus einem KMS-Schlüssel. Um die zu löschenden Tags zu identifizieren, geben Sie die Tag-Schlüssel an. Sie können diese Operation nicht verwenden, um Tags aus KMS-Schlüsseln in einem anderen AWS-Konto zu löschen.

Wenn sie erfolgreich ist, gibt die `UntagResource`-Operation keine Ausgabe zurück. Wenn der angegebene Tag-Schlüssel nicht auf dem KMS-Schlüssel gefunden wird, wird keine Ausnahme ausgelöst und keine Antwort zurückgegeben. Um zu bestätigen, dass der Vorgang funktioniert hat, verwenden Sie den ListResourceTags Vorgang .

Dieser Befehl löscht beispielsweise das **`Purpose`**-Tag und alle zugehörigen Werte von dem angegebenen KMS-Schlüssel.

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys
 Purpose
```

Steuern des Zugriffs auf Tags

Zum Hinzufügen, Anzeigen und Löschen von Tags, entweder in der AWS KMS-Konsole oder mithilfe der API, benötigen Prinzipale Markierungs-Berechtigungen. Sie können diese Berechtigungen in Schlüsselrichtlinien bereitstellen. Sie können sie auch in IAM-Richtlinien (einschließlich VPC-Endpunktrichtlinien) bereitstellen, aber nur wenn die Schlüsselrichtlinie es erlaubt. Die AWSKeyManagementServicePowerUser von verwaltete Richtlinie ermöglicht es Prinzipalen, Tags auf allen KMS-Schlüsseln, auf die das Konto zugreifen kann, zu markieren, aufzuheben und aufzulisten.

Sie können diese Berechtigungen auch einschränken, indem Sie globale AWS-Bedingungsschlüssel für Tags verwenden. In können AWS KMSdiese Bedingungen den Zugriff auf Tagging-Operationen wie TagResource und steuernUntagResource.

> **Note**
>
> Seien Sie vorsichtig, wenn Sie Prinzipalen die Berechtigung zum Verwalten von Tags und Aliasen erteilen. Wenn Sie eine Markierung oder einen Alias ändern, wird dadurch die Berechtigung für den kundenverwalteten Schlüssel erteilt oder verweigert. Details dazu finden

Sie unter ABAC für AWS KMS und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.

Beispielrichtlinien und weitere Informationen finden Sie unter Zugriffssteuerung anhand von Tag-Schlüsseln im IAM-Benutzerhandbuch.

Berechtigungen zum Erstellen und Verwalten von Tags funktionieren wie folgt.

kms:TagResource

Erlaubt es Prinzipalen, Tags hinzuzufügen oder zu bearbeiten. Um Tags beim Erstellen eines KMS-Schlüssels hinzuzufügen, muss der Prinzipal über die Berechtigung in einer IAM-Richtlinie verfügen, die nicht auf bestimmte KMS-Schlüssel beschränkt ist.

kms:ListResourceTags

Erlaubt es Prinzipalen, Tags auf KMS-Schlüsseln anzuzeigen.

kms:UntagResource

Erlaubt es Prinzipalen, Tags aus KMS-Schlüsseln zu löschen.

Tag-Berechtigungen in Richtlinien

Sie können Markierungs-Berechtigungen in einer Schlüsselrichtlinie oder einer IAM-Richtlinie bereitstellen. Beispielsweise erteilt die folgende Beispiel-Schlüsselrichtlinie ausgewählten Benutzern Markierungs-Berechtigung für den KMS-Schlüssel. Sie erteilt allen Benutzern, die die Beispiel-Administrator- oder Entwicklerrollen übernehmen können, die Berechtigung zum Anzeigen von Tags.

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

Um Prinzipalen Markierungs-Berechtigung für mehrere KMS-Schlüssel zu erteilen, können Sie eine IAM-Richtlinie verwenden. Damit diese Richtlinie wirksam wird, muss die Schlüsselrichtlinie für jeden KMS-Schlüssel dem Konto erlauben, mithilfe von IAM-Richtlinien den Zugriff auf den KMS-Schlüssel zu steuern.

Beispielsweise kann die folgende IAM-Richtlinie den Prinzipalen erlauben, KMS-Schlüssel zu erstellen. Außerdem erlaubt es ihnen, Tags für alle KMS-Schlüssel im angegebenen Konto zu erstellen und zu verwalten. Diese Kombination ermöglicht es den Prinzipalen, den Tags-Parameter der -CreateKeyOperation zu verwenden, um einem KMS-Schlüssel Tags hinzuzufügen, während sie ihn erstellen.

```
{
  "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/*"
    }
  ]
}
```

Beschränken von Tag-Berechtigungen

Sie können Markierungs-Berechtigungen einschränken, indem Sie Richtlinienbedingungen verwenden. Die folgenden Richtlinienbedingungen können auf die `kms:TagResource`- und `kms:UntagResource`-Berechtigungen angewendet werden. Beispielsweise können Sie mit der `aws:RequestTag/`*`tag-key`*-Bedingung einem Prinzipal erlauben, nur bestimmte Tags hinzuzufügen, oder verhindern, dass ein Prinzipal Tags mit bestimmten Tag-Schlüsseln hinzufügen kann. Alternativ können Sie auch die `kms:KeyOrigin`-Bedingung verwenden, um zu verhindern, dass Prinzipale KMS-Schlüssel mit importiertem Schlüsselmaterial markieren oder entmarkieren.

- aws:RequestTag
- aws:ResourceTag/tag-key (nur IAM-Richtlinien)
- aws:TagKeys
- kms:CallerAccount
- kms:KeySpec
- kms:KeyUsage
- kms:KeyOrigin
- kms:ViaService

Eine bewährte Methode bei der Verwendung von Tags zum Steuern des Zugriffs auf KMS-Schlüssel ist die Verwendung der Bedingungsschlüssel `aws:RequestTag/`*`tag-key`*oder `aws:TagKeys`, um zu bestimmen, welche Tags (oder Tag-Schlüssel) erlaubt sind.

Die folgende IAM-Richtlinie ähnelt beispielsweise der vorherigen. Diese Richtlinie erlaubt den Prinzipalen jedoch das Erstellen von Tags (`TagResource`) und das Löschen von Tags (`UntagResource`) nur für Tags mit einem `Project`-Tag-Schlüssel.

Da - `TagResource` und -`UntagResource`Anforderungen mehrere Tags enthalten können, müssen Sie einen - `ForAllValues` oder -`ForAnyValue`Satzoperator mit der Bedingung aws:TagKeys angeben. Der `ForAnyValue`-Operator erfordert, dass mindestens einer der Tag-Schlüssel in der Anforderung mit einem der Tag-Schlüssel in der Richtlinie übereinstimmen muss. Der `ForAllValues`-Operator erfordert, dass alle der Tag-Schlüssel in der Anforderung mit einem der Tag-Schlüssel in der Richtlinie übereinstimmen müssen. Der `ForAllValues` Operator gibt auch zurück, `true` wenn die Anforderung keine Tags enthält, aber TagResource und UntagResource schlagen fehl, wenn keine Tags angegeben sind. Ausführliche Informationen zu den Satz-Operatoren finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch.

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

```
}
```

Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel

Sie können den Zugriff auf AWS KMS keys anhand der Tags auf dem KMS-Schlüssel steuern. Sie können beispielsweise eine IAM-Richtlinie schreiben, die es Prinzipalen nur erlaubt, die KMS-Schlüssel mit einem bestimmten Tag zu aktivieren und zu deaktivieren. Oder Sie können eine IAM-Richtlinie verwenden, um zu verhindern, dass Prinzipale KMS-Schlüssel in kryptografischen Operationen verwenden, es sei denn, der KMS-Schlüssel hat ein bestimmtes Tag.

Dieses Feature ist Teil der AWS KMS-Unterstützung für attributbasierte Zugriffssteuerung (ABAC). Weitere Informationen zur Verwendung von Tags, um den Zugriff auf Ihre AWS-Ressourcen zu steuern, finden Sie unter Was ist ABAC für AWS? und Steuerung des Zugriffs auf AWS-Ressourcen mit Ressourcen-Tags im IAM-Benutzerhandbuch. Hilfe zur Behebung von Zugriffsproblemen im Zusammenhang mit ABAC finden Sie unter Fehlerbehebung bei ABAC für AWS KMS.

> **Note**
>
> Es kann bis zu fünf Minuten dauern, bis Tag- und Alias-Änderungen Auswirkungen auf die KMS-Schlüsselautorisierung haben. Letzte Änderungen sind möglicherweise in API-Operationen sichtbar, bevor sie sich auf die Autorisierung auswirken.

AWS KMS unterstützt den globalen Bedingungskontextschlüssel aws:ResourceTag/tag-key , mit dem Sie den Zugriff auf KMS-Schlüssel basierend auf den Tags auf dem KMS-Schlüssel steuern können. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html Da mehrere KMS-Schlüssel das gleiche Tag haben können, können Sie mit diesem Feature die Berechtigung auf einen ausgewählten Satz von KMS-Schlüsseln anwenden. Sie können die KMS-Schlüssel im Satz auch einfach ändern, indem Sie deren Tags ändern.

In AWS KMS wird der Bedingungsschlüssel `aws:ResourceTag/`*`tag-key`* nur in IAM-Richtlinien unterstützt. Es wird nicht in Schlüsselrichtlinien unterstützt, die nur für einen KMS-Schlüssel gelten, oder für Operationen, die keinen bestimmten KMS-Schlüssel verwenden, z. B. die - ListKeys oder -ListAliasesOperationen.

Die Steuerung des Zugriffs mit Tags bietet eine einfache, skalierbare und flexible Möglichkeit, Berechtigungen zu verwalten. Wenn es jedoch nicht richtig gestaltet und verwaltet wird, kann es versehentlich den Zugriff auf Ihre KMS-Schlüssel zulassen oder verweigern. Wenn Sie Tags verwenden, um den Zugriff zu steuern, sollten Sie die folgenden Methoden berücksichtigen.

- Verwenden Sie Tags, um beim Zugriff die bewährte Methode der geringsten Berechtigung zu befolgen. Geben Sie IAM-Prinzipalen nur die Berechtigungen ein, die sie nur für die KMS-Schlüssel benötigen, die sie verwenden oder verwalten müssen. Verwenden Sie beispielsweise Tags, um die KMS-Schlüssel zu markieren, die für ein Projekt verwendet werden. Geben Sie dann dem Projektteam die Berechtigung, nur KMS-Schlüssel mit dem Projekt-Tag zu verwenden.
- Seien Sie vorsichtig, wenn Sie Prinzipalen die `kms:TagResource`- und `kms:UntagResource`-Berechtigung erteilen, mit denen sie Tags hinzufügen, bearbeiten und löschen können. Wenn Sie Tags verwenden, um den Zugriff auf KMS-Schlüssel zu steuern, kann das Ändern eines Tags Prinzipalen die Berechtigung zur Verwendung von KMS-Schlüsseln erteilen, die andernfalls nicht über die Berechtigung verfügen. Es kann auch den Zugriff auf KMS-Schlüssel verweigern, die andere Prinzipale für ihre Aufträge benötigen. Schlüsseladministratoren, die nicht über die Berechtigung zum Ändern von Schlüsselrichtlinien oder zum Erstellen von Erteilungen verfügen, können den Zugriff auf KMS-Schlüssel steuern, wenn sie über die Berechtigung zum Verwalten von Tags verfügen.

  Verwenden Sie nach Möglichkeit eine Richtlinienbedingung, z. B. `aws:RequestTag/`*`tag-key`* oder `aws:TagKeys` zum Beschränken der Markierungs-Berechtigungen eines Prinzipals auf bestimmte Tags oder Tag-Muster für bestimmte KMS-Schlüssel.
- Überprüfen Sie die Prinzipale in Ihrem AWS-Konto, die derzeit über Markierungs- und Entmarkierungs-Berechtigungen verfügen, und passen Sie sie gegebenenfalls an. Zum Beispiel enthält die Standard-Schlüsselrichtlinie für Schlüsseladministratoren der Konsole die `kms:TagResource`- und `kms:UntagResource`-Berechtigung für diesen KMS-Schlüssel. IAM-Richtlinien erlauben vielleicht Markierungs- und Entmarkierungs-Berechtigungen für alle KMS-Schlüssel. Die von AWSKeyManagementServicePowerUser verwaltete Richtlinie erlaubt es beispielsweise Prinzipalen, Tags auf allen KMS-Schlüsseln zu markieren, zu entmarkieren und aufzulisten.
- Bevor Sie eine Richtlinie festlegen, die von einem Tag abhängt, überprüfen Sie die Tags der KMS-Schlüssel in Ihrem AWS-Konto. Stellen Sie sicher, dass Ihre Richtlinie nur für die Tags gilt, die Sie einschließen möchten. Verwenden Sie CloudTrail Protokolle und CloudWatch Alarme, um Sie auf Tag-Änderungen aufmerksam zu machen, die sich auf den Zugriff auf Ihre KMS-Schlüssel auswirken könnten.
- Die tag-basierten Richtlinienbedingungen verwenden Musterabgleich; sie sind nicht an eine bestimmte Instance eines Tags gebunden. Eine Richtlinie, die Tag-basierte Bedingungsschlüssel verwendet, wirkt sich auf alle neuen und vorhandenen Tags aus, die dem Muster entsprechen. Wenn Sie ein Tag löschen und neu erstellen, das einer Richtlinienbedingung entspricht, gilt die Bedingung für das neue Tag, genau wie für das alte Tag.

Betrachten Sie beispielsweise die folgende IAM-Richtlinie: Sie ermöglicht es den Prinzipalen, die Entschlüsselungsvorgänge GenerateDataKeyWithoutPlaintext und nur für KMS-Schlüssel in Ihrem Konto aufzurufen, die die Region Asien-Pazifik (Singapur) sind und ein `"Project"="Alpha"` Tag haben. Sie können diese Richtlinie an Rollen im Beispiel Alpha-Projekt anfügen.

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

Die folgende IAM-Beispielrichtlinie erlaubt es den Prinzipalen, den KMS-Schlüssel im Konto für kryptografische Operationen zu verwenden. Sie verbietet es den Prinzipalen jedoch, diese kryptografischen Operationen für KMS-Schlüssel mit einem `"Type"="Reserved"`-Tag oder keinem `"Type"`-Tag zu verwenden.

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

Aktivieren und Deaktivieren von Schlüsseln

Sie können kundenverwaltete Schlüssel aktivieren und deaktivieren. Beim Erstellen eines KMS-Schlüssels ist er standardmäßig aktiviert. Wenn Sie einen KMS-Schlüssel deaktivieren, kann er in keiner kryptografischen Produktion verwendet werden, bis Sie die Option erneut aktivieren.

Da es vorübergehend ist und leicht rückgängig gemacht werden kann, ist das Deaktivieren eines KMS-Schlüssels eine sichere Alternative zum Löschen eines KMS-Schlüssels, einer Aktion, die endgültig und irreversibel ist. Wenn Sie erwägen, einen KMS-Schlüssel zu löschen, deaktivieren Sie ihn zuerst und legen Sie einen CloudWatch Alarm oder einen ähnlichen Mechanismus fest, um sicherzustellen, dass Sie den Schlüssel niemals zum Entschlüsseln verschlüsselter Daten verwenden müssen.

Wenn Sie einen KMS-Schlüssel deaktivieren, wird er sofort unbrauchbar (je nach letztendlicher Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel geschützten Datenschlüsseln verschlüsselt wurden, sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, von denen viele Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Sie können Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel nicht aktivieren oder deaktivieren. Von AWS verwaltete Schlüssel sind dauerhaft für die Verwendung durch -Services aktiviert, die AWS KMS verwenden. AWS-eigene Schlüssel werden ausschließlich von dem Service verwaltet, dem sie gehören.

> **Note**
>
> AWS KMS dreht nicht die das Schlüsselmaterial von kundenverwalteten Schlüsseln, solange diese deaktiviert sind. Weitere Informationen finden Sie unter So funktioniert die Schlüsselrotation.

Themen

- Aktivieren und Deaktivieren von KMS-Schlüsseln (Konsole)
- Aktivieren und Deaktivieren von KMS-Schlüsseln (AWS KMS-API)

Aktivieren und Deaktivieren von KMS-Schlüsseln (Konsole)

Sie können die AWS KMS-Konsole verwenden, um kundenverwaltete Schlüssel zu aktivieren und zu deaktivieren.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.

2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Aktivieren Sie das Kontrollkästchen für die KMS-Schlüssel, die Sie aktivieren oder deaktivieren möchten.
5. Um einen KMS-Schlüssel zu aktivieren, wählen Sie Key actions (Schlüsselaktionen) und Enable (aktivieren) aus. Um einen KMS-Schlüssel zu deaktivieren, wählen Sie Key actions (Schlüsselaktionen) und Disable (deaktivieren) aus.

Aktivieren und Deaktivieren von KMS-Schlüsseln (AWS KMS-API)

Die -EnableKeyOperation aktiviert eine deaktivierte AWS KMS key. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen. Der Parameter `key-id` muss angegeben werden.

Diese Produktion gibt keine Ausgabe zurück. Um den Schlüsselstatus anzuzeigen, verwenden Sie die -DescribeKeyOperation.

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

Die -DisableKeyOperation deaktiviert einen aktivierten KMS-Schlüssel. Der Parameter `key-id` muss angegeben werden.

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

Diese Produktion gibt keine Ausgabe zurück. Um den Schlüsselstatus anzuzeigen, verwenden Sie die -DescribeKeyOperation und sehen Sie sich das `Enabled` Feld an.

```
$ 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"
        ]
    }
}
```

Rotierend AWS KMS keys

Um neue kryptografische Daten für Ihre kundenverwalteten Schlüssel zu erstellen, können Sie neue KMS-Schlüssel erstellen und anschließend Ihre Anwendungen oder Aliasse so ändern, dass diese die neuen KMS-Schlüssel verwenden. Oder Sie können das Schlüsselmaterial, das einem vorhandenen KMS-Schlüssel zugeordnet ist, rotieren, indem Sie die automatische Schlüsselrotation aktivieren oder eine Rotation bei Bedarf durchführen.

Wenn Sie die automatische Schlüsselrotation für einen KMS-Schlüssel aktivieren, AWS KMS wird standardmäßig jedes Jahr neues kryptografisches Material für den KMS-Schlüssel generiert. Sie können auch einen benutzerdefinierten Wert angebenrotation-period, um die Anzahl der Tage nach der Aktivierung der automatischen Schlüsselrotation zu definieren, bei der das Schlüsselmaterial rotiert AWS KMS wird, sowie die Anzahl der Tage zwischen den einzelnen automatischen Rotationen danach. Wenn Sie die Schlüsselrotation sofort einleiten müssen, können Sie die Rotation bei Bedarf durchführen, unabhängig davon, ob die automatische Schlüsselrotation aktiviert ist oder nicht. Rotationen auf Anforderung ändern keine bestehenden Zeitpläne für die automatische Rotation.

AWS KMS speichert alle früheren Versionen des kryptografischen Materials auf unbestimmte Zeit, sodass Sie alle mit diesem KMS-Schlüssel verschlüsselten Daten entschlüsseln können. AWS KMS löscht kein rotiertes Schlüsselmaterial, bis Sie den KMS-Schlüssel löschen. Sie können die Rotation des Schlüsselmaterials für Ihre KMS-Schlüssel in Amazon CloudWatch und in der AWS Key Management Service Konsole verfolgen. AWS CloudTrail Mithilfe von GetKeyRotationStatusOperation können Sie auch überprüfen, ob die automatische Rotation für einen KMS-Schlüssel aktiviert ist, und alle laufenden On-Demand-Rotationen identifizieren. Mithilfe von ListKeyRotationsOperation können Sie sich die Details abgeschlossener Rotationen anzeigen lassen.

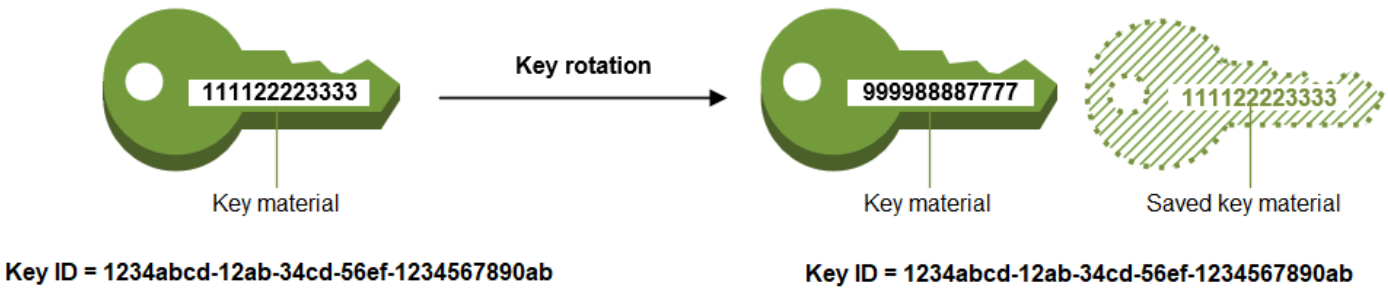
Wenn Sie einen rotierten KMS-Schlüssel zum Verschlüsseln von Daten verwenden, AWS KMS wird das aktuelle Schlüsselmaterial verwendet. Wenn Sie den rotierten KMS-Schlüssel zum Entschlüsseln von Chiffretext verwenden, AWS KMS wird die Version des Schlüsselmaterials verwendet, mit der er verschlüsselt wurde. Sie können keine bestimmte Version des Schlüsselmaterials für Entschlüsselungsvorgänge auswählen. Es AWS KMS wird automatisch die richtige Version ausgewählt. Da mit dem entsprechenden Schlüsselmaterial AWS KMS transparent entschlüsselt wird, können Sie einen rotierten KMS-Schlüssel problemlos in Anwendungen und AWS-Services ohne Codeänderungen verwenden.

Die automatische Schlüsseldrehung hat jedoch keine Auswirkungen auf die Daten, die der KMS-Schlüssel schützt. Es dreht nicht die Datenschlüssel, die der KMS-Schlüssel generiert hat und verschlüsselt keine Daten neu, die durch den KMS-Schlüssel geschützt sind. Es minimiert auch nicht die Auswirkungen eines kompromittierten Datenschlüssels.

AWS KMS unterstützt die automatische und bedarfsgesteuerte Schlüsselrotation nur bei symmetrischer Verschlüsselung von KMS-Schlüsseln mit Schlüsselmaterial, das erstellt. AWS KMS Die automatische Drehung ist optional für vom Kunden verwaltete KMS-Schlüssel. AWS KMS dreht immer jedes Jahr das Schlüsselmaterial für AWS -verwaltete KMS-Schlüssel. Die Rotation AWS eigener KMS-Schlüssel wird von dem AWS Dienst verwaltet, dem der Schlüssel gehört.

> **Note**
>
> Der Rotationszeitraum für Von AWS verwaltete Schlüssel hat sich im Mai 2022 geändert. Details hierzu finden Sie unter Von AWS verwaltete Schlüssel.

Durch die Schlüsseldrehung wird nur das Schlüsselmaterial des CMK geändert. Dabei handelt es sich um das kryptografische Geheimnis, das für Verschlüsselungsoperationen verwendet werden. Der KMS-Schlüssel ist dieselbe logische Ressource, unabhängig davon, ob oder wie oft das Schlüsselmaterial geändert wird. Die Eigenschaften des KMS-Schlüssels werden nicht geändert, wie in der folgenden Abbildung dargestellt.



Möglicherweise möchten Sie einen neuen KMS-Schlüssel erstellen und anstelle des ursprünglichen KMS-Schlüssels verwenden. Dies hat den gleichen Effekt wie das Drehen des Schlüsselmaterials in einem vorhandenen KMS-Schlüssel, sodass es häufig als manuelle Drehung des Schlüssels betrachtet wird. Die manuelle Rotation ist eine gute Wahl, wenn Sie KMS-Schlüssel rotieren möchten, die nicht für eine automatische Schlüsselrotation in Frage kommen, einschließlich asymmetrischer KMS-Schlüssel, HMAC-KMS-Schlüssel, KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern und KMS-Schlüssel mit importiertem Schlüsselmaterial.

Schlüsselrotation und Preise

AWS KMS berechnet eine monatliche Gebühr für die erste und zweite Rotation des für Ihren KMS-Schlüssel verwalteten Schlüsselmaterials. Diese Preiserhöhung ist bei der zweiten Rotation begrenzt, und alle nachfolgenden Rotationen werden nicht in Rechnung gestellt. Für Einzelheiten vgl. AWS Key Management Service -Preise.

> **Note**
>
> Sie können den AWS Cost Explorer Service verwenden, um eine Aufschlüsselung Ihrer Gebühren für die Aufbewahrung Ihrer Schlüssel einzusehen. Sie können Ihre Ansicht beispielsweise so filtern, dass die Gesamtkosten für Schlüssel angezeigt werden, die als aktuelle und rotierte KMS-Schlüssel abgerechnet werden, indem Sie den `$REGION-KMS-Keys` als den Nutzungstyp angeben und die Daten nach API-Vorgang gruppieren. Möglicherweise werden Ihnen immer noch Instances des älteren `Unknown`-API-Vorgangs für historische Daten angezeigt.

Schlüsselrotation und Kontingente

Jeder KMS-Schlüssel zählt als Schlüssel bei der Berechnung wichtiger Ressourcenkontingente, unabhängig von der Anzahl der gedrehten Schlüsselmaterialversionen.

Weitere Informationen zu Schlüsselmaterial und Drehung finden Sie unter AWS Key Management Service Kryptografische Details.

Themen

- Warum sollten KMS-Schlüssel rotiert werden?
- So funktioniert die Schlüsselrotation
- So aktivieren und deaktivieren Sie die automatische Schlüsselrotation:

- Wie führe ich eine Schlüsselrotation bei Bedarf durch
- Manuelles Rotieren von Schlüsseln

Warum sollten KMS-Schlüssel rotiert werden?

Bewährte kryptografische Verfahren verhindern die umfassende Wiederverwendung von Schlüsseln, die Daten direkt verschlüsseln, wie z. B. generierte Datenschlüssel. AWS KMS Wenn 256-Bit-Datenschlüssel Millionen von Nachrichten verschlüsseln, können sie erschöpft sein und anfangen, Geheimtext mit subtilen Mustern zu erzeugen, den clevere Akteure ausnutzen können, um die Bits im Schlüssel zu entdecken. Um diese Erschöpfung der Schlüssel zu vermeiden, empfiehlt es sich, Datenschlüssel einmal oder nur ein paar Mal zu verwenden, wodurch das Schlüsselmaterial effektiv rotiert wird.

KMS-Schlüssel werden jedoch am häufigsten als Mantelschlüssel verwendet, die auch als Schlüssel zur Schlüsselverschlüsselung bezeichnet werden. Anstatt Daten zu verschlüsseln, verschlüsseln Mantelschlüssel die Datenschlüssel, die Ihre Daten verschlüsseln. Daher werden sie weitaus seltener verwendet als Datenschlüssel und werden fast nie so oft wiederverwendet, dass das Risiko besteht, dass die Schlüssel erschöpft werden.

Trotz dieses sehr geringen Erschöpfungsrisikos müssen Sie Ihre KMS-Schlüssel möglicherweise aufgrund von Geschäfts- oder Vertragsregeln oder behördlichen Vorschriften rotieren. Wenn Sie gezwungen sind, KMS-Schlüssel zu rotieren, empfehlen wir, die automatische Schlüsselrotation zu verwenden, sofern sie unterstützt wird, und die manuelle Schlüsselrotation, wenn die automatische Schlüsselrotation nicht unterstützt wird.

Sie könnten erwägen, Rotationen auf Abruf durchzuführen, um wichtige Funktionen der Materialrotation zu demonstrieren oder Automatisierungsskripte zu validieren. Wir empfehlen, für ungeplante Rotationen On-Demand-Rotationen und, wann immer möglich, die automatische Schlüsselrotation mit einem benutzerdefinierten Rotationsperiode zu verwenden.

So funktioniert die Schlüsselrotation

Die Schlüsselrotation in AWS KMS ist so konzipiert, dass sie transparent und einfach zu bedienen ist. AWS KMS unterstützt die optionale automatische und bedarfsgesteuerte Schlüsselrotation nur für vom Kunden verwaltete Schlüssel.

Automatische Schlüsselrotation

AWS KMS rotiert den KMS-Schlüssel automatisch am nächsten Rotationsdatum, das durch Ihren Rotationsperiode definiert ist. Sie müssen das Update nicht selbst planen.

Rotation auf Anfrage

Initiieren Sie sofort die Rotation des mit Ihrem KMS-Schlüssel verknüpften Schlüsselmaterials, unabhängig davon, ob die automatische Schlüsselrotation aktiviert ist oder nicht.

Verwalten von Schlüsselmaterial

AWS KMS behält das gesamte Schlüsselmaterial für einen KMS-Schlüssel bei, auch wenn die Schlüsselrotation deaktiviert ist. AWS KMS löscht Schlüsselmaterial nur, wenn Sie den KMS-Schlüssel löschen.

Verwenden von Schlüsselmaterial

Wenn Sie einen rotierten KMS-Schlüssel zum Verschlüsseln von Daten verwenden, AWS KMS verwendet das aktuelle Schlüsselmaterial. Wenn Sie mit dem KMS-Schlüssel Daten entschlüsseln, verwendet AWS KMS das Schlüsselmaterial, das zum Verschlüsseln verwendet wurde. Sie können keine bestimmte Version des Schlüsselmaterials für Entschlüsselungsvorgänge auswählen. Es AWS KMS wird automatisch die richtige Version ausgewählt.

Zeitraum der Rotation

Der Rotationszeitraum definiert die Anzahl der Tage nach der Aktivierung der automatischen Schlüsselrotation, wodurch Ihr Schlüsselmaterial rotiert AWS KMS wird, und die Anzahl der Tage zwischen den einzelnen automatischen Schlüsselrotationen danach. Wenn Sie keinen Wert für `RotationPeriodInDays` die Aktivierung der automatischen Schlüsselrotation angeben, ist der Standardwert 365 Tage.

Sie können den RotationPeriodInDays Bedingungsschlüssel kms: verwenden, um die Werte, die Prinzipale im Parameter angeben können, weiter einzuschränken. `RotationPeriodInDays`

Rotationsdatum

AWS KMS rotiert den KMS-Schlüssel automatisch an dem durch Ihre Rotationsperiode definierten Rotationsdatum. Der Standardrotationszeitraum beträgt 365 Tage.

Kundenverwaltete Schlüssel

Da die automatische Schlüsselrotation bei vom Kunden verwalteten Schlüsseln optional ist und jederzeit aktiviert und deaktiviert werden kann, hängt das Rotationsdatum von dem

Datum ab, an dem die Rotation zuletzt aktiviert wurde. Das Datum kann sich ändern, wenn Sie den Rotationszeitraum für einen Schlüssel ändern, für den Sie zuvor die automatische Schlüsselrotation aktiviert haben. Das Rotationsdatum kann sich im Laufe der Lebensdauer des Schlüssels mehrfach ändern.

Wenn Sie beispielsweise am 1. Januar 2022 einen vom Kunden verwalteten Schlüssel erstellen und die automatische Schlüsselrotation mit dem AWS KMS Standardrotationszeitraum von 365 Tagen am 15. März 2022 aktivieren, wird das Schlüsselmaterial am 15. März 2023, 15. März 2024 und danach alle 365 Tage rotiert.

In den folgenden Beispielen wird davon ausgegangen, dass die automatische Schlüsselrotation mit dem Standardrotationszeitraum von 365 Tagen aktiviert wurde. Diese Beispiele zeigen Sonderfälle, die sich auf die Rotationsperiode eines Schlüssels auswirken können.

- Schlüsselrotation deaktivieren – Wenn Sie zu einem beliebigen Zeitpunkt die automatische Schlüsselrotation deaktivieren, verwendet der KMS-Schlüssel weiterhin die Version des Schlüsselmaterials, das er verwendet hat, als die Rotation deaktiviert wurde. Wenn Sie die automatische Schlüsselrotation wieder aktivieren, AWS KMS rotiert das Schlüsselmaterial auf der Grundlage des neuen Aktivierungsdatums für die Rotation.
- Deaktivierte KMS-Schlüssel — Solange ein KMS-Schlüssel deaktiviert ist, wird AWS KMS er nicht rotiert. Der Status der Schlüsseldrehung ändert sich jedoch nicht, und Sie können ihn nicht ändern, solange der KMS-Schlüssel deaktiviert ist. Wenn der KMS-Schlüssel wieder aktiviert wird und das Schlüsselmaterial sein letztes geplantes Rotationsdatum überschritten hat, wird es AWS KMS sofort rotiert. Wenn das Schlüsselmaterial seinen letzten geplanten Rotationstermin nicht verpasst hat, AWS KMS wird der ursprüngliche Zeitplan für die Schlüsselrotation wieder aufgenommen.
- KMS-Schlüssel, deren Löschung noch aussteht — Solange ein KMS-Schlüssel noch gelöscht werden AWS KMS muss, wird er nicht rotiert. Als Status der Schlüsselrotation wird `false` eingestellt und er kann von Ihnen nicht mehr geändert werden, während der Löschvorgang ansteht. Wenn der Löschvorgang abgebrochen wird, wird der vorherige Status der Schlüsselrotation wiederhergestellt. Wenn das Schlüsselmaterial seinen letzten geplanten Rotationstermin überschritten hat, wird es sofort AWS KMS rotiert. Wenn das Schlüsselmaterial seinen letzten geplanten Rotationstermin nicht verpasst hat, AWS KMS wird der ursprüngliche Zeitplan für die Schlüsselrotation wieder aufgenommen.

Von AWS verwaltete Schlüssel

AWS KMS wechselt automatisch Von AWS verwaltete Schlüssel jedes Jahr (ungefähr 365 Tage). Sie können die Schlüsseldrehung von Von AWS verwaltete Schlüssel nicht aktivieren oder deaktivieren.

Das Schlüsselmaterial für eine Von AWS verwalteter Schlüssel wird zunächst ein Jahr nach dem Erstellungsdatum rotiert und danach jedes Jahr (ungefähr 365 Tage nach der letzten Rotation).

> **Note**
>
> Im Mai 2022 AWS KMS wurde der Rotationsplan Von AWS verwaltete Schlüssel von allen drei Jahren (etwa 1.095 Tage) auf jedes Jahr (ungefähr 365 Tage) geändert.
> Neue Von AWS verwaltete Schlüssel werden automatisch ein Jahr nach ihrer Erstellung und danach ungefähr jedes Jahr rotiert.
> Bestehende Von AWS verwaltete Schlüssel werden automatisch ein Jahr nach ihrer letzten Rotation und danach jedes Jahr gewechselt.

AWS-eigene Schlüssel

Sie können die Schlüsseldrehung von AWS-eigene Schlüssel nicht aktivieren oder deaktivieren. Die Strategie der Schlüsselrotation für einen AWS-eigener Schlüssel wird durch den AWS Dienst bestimmt, der den Schlüssel erstellt und verwaltet. Weitere Informationen finden Sie im Benutzerhandbuch oder Entwicklerhandbuch für den Service unter dem Thema Verschlüsselung im Ruhezustand.

Unterstützte KMS-Schlüsseltypen

Die automatische Schlüsseldrehung wird nur für KMS-Schlüssel mit symmetrischer Verschlüsselung mit Schlüsselmaterial, das AWS KMS generiert (Ursprung = AWS_KMS).

Die automatische Schlüsseldrehung wird bei den folgenden Typen von KMS-Schlüsseln nicht unterstützt. Sie können diese KMS-Schlüssel jedoch manuell drehen.

- Asymmetrische KMS-Schlüssel
- HMAC-KMS-Schlüssel
- KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern

- So zeigen Sie KMS-Schlüssel mit importiertem Schlüsselmaterial an

Multiregionale Schlüssel

Sie können die automatische Drehung des Schlüsselmaterials für multiregionale Schlüssel aktivieren und deaktivieren. Sie legen die Eigenschaft nur für den Primärschlüssel fest. Wenn die Schlüssel AWS KMS synchronisiert werden, kopiert er die Eigenschaftseinstellung aus dem Primärschlüssel in seine Replikatschlüssel. Wenn das Schlüsselmaterial des Primärschlüssels rotiert wird, wird dieses Schlüsselmaterial AWS KMS automatisch in alle zugehörigen Replikatschlüssel kopiert. Details hierzu finden Sie unter Drehen von multiregionalen Schlüsseln.

AWS Dienste

Sie können für die kundenverwalteten KMS-Schlüssel, die Sie für die serverseitige Verschlüsselung in AWS -Services verwenden, eine automatische Schlüsseldrehung aktivieren. Die jährliche Drehung ist transparent und mit AWS -Services kompatibel.

Überwachen der Schlüsselrotation.

Wenn das Schlüsselmaterial für einen Von AWS verwalteter Schlüsseloder vom Kunden verwalteten Schlüssel AWS KMS rotiert wird, schreibt es ein `KMS CMK Rotation` Ereignis an Amazon EventBridge und ein RotateKey Ereignis in Ihr AWS CloudTrail Protokoll. Sie können die diese Datensätze verwenden, um zu überprüfen, ob der KMS-Schlüssel gedreht wurde.

In der AWS Key Management Service Konsole können Sie sich die Anzahl der verbleibenden On-Demand-Rotationen und eine Liste aller abgeschlossenen Schlüsselmaterialrotationen für einen KMS-Schlüssel anzeigen lassen.

Sie können den ListKeyRotationsVorgang verwenden, um die Details der abgeschlossenen Rotationen einzusehen.

Letztendliche Datenkonsistenz

Die Schlüsselrotation unterliegt letztlich den gleichen Konsistenzeffekten wie andere AWS KMS Verwaltungsvorgänge. Es kann zu einer leichten Verzögerung kommen, bevor das neue Schlüsselmaterial in allen Bereichen von AWS KMS verfügbar ist. Das Drehen von Schlüsselmaterial verursacht jedoch keine Unterbrechung oder Verzögerung kryptografischer Operationen. Das aktuelle Schlüsselmaterial wird in kryptografischen Operationen verwendet, bis das neue Schlüsselmaterial überall in AWS KMS verfügbar ist. Wenn das Schlüsselmaterial für einen Schlüssel mit mehreren Regionen automatisch rotiert wird, wird das aktuelle Schlüsselmaterial AWS KMS verwendet, bis das neue Schlüsselmaterial in allen Regionen mit einem zugehörigen Schlüssel für mehrere Regionen verfügbar ist.

So aktivieren und deaktivieren Sie die automatische Schlüsselrotation:

Wenn Sie die automatische Schlüsselrotation für einen KMS-Schlüssel aktivieren, AWS KMS generiert standardmäßig jedes Jahr neues kryptografisches Material für den KMS-Schlüssel. Sie können auch einen benutzerdefinierten Wert angebenrotation-period, um die Anzahl der Tage nach der Aktivierung der automatischen Schlüsselrotation zu definieren, bei der das Schlüsselmaterial rotiert AWS KMS wird, sowie die Anzahl der Tage zwischen den einzelnen automatischen Rotationen danach.

Die automatische Schlüsselrotation bietet folgende Vorteile:

- Die Eigenschaften des KMS-Schlüssels, einschließlich Schlüssel-ID, Schlüssel-ARN, Region, Richtlinien und Berechtigungen, werden bei der Drehung des Schlüssels nicht geändert.
- Sie müssen keine Anwendungen oder Aliasse ändern, die sich auf die KMS-Schlüssel-ID oder den ARN beziehen.
- Drehendes Schlüsselmaterial beeinträchtigt die Verwendung des KMS-Schlüssels in keinem AWS-Service.
- Nachdem Sie die Schlüsselrotation aktiviert haben, wird der KMS-Schlüssel automatisch am nächsten Rotationsdatum AWS KMS rotiert, das durch Ihre Rotationsperiode definiert wird. Sie müssen das Update nicht selbst planen.

Autorisierte Benutzer können die AWS KMS Konsole und die AWS KMS API verwenden, um die automatische Schlüsselrotation zu aktivieren und zu deaktivieren und den Status der Schlüsselrotation einzusehen.

Themen

- Automatische Schlüsselrotation aktivieren und deaktivieren (Konsole)
- Automatische Schlüsselrotation (AWS KMS API) aktivieren und deaktivieren

Automatische Schlüsselrotation aktivieren und deaktivieren (Konsole)

1. Melden Sie sich bei der Konsole AWS Key Management Service (AWS KMS) unter https://console.aws.amazon.com/kms an AWS Management Console und öffnen Sie sie.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.

3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. (Sie können die Drehung von Von AWS verwaltete Schlüssel nicht aktiviern oder deaktivieren. Diese werden automatisch jedes Jahr gedreht.)
4. Wählen Sie den Alias oder die Schlüssel-ID eines KMS-Schlüssels.
5. Wählen Sie die Registerkarte Key rotation (Schlüsselrotation).

   Die Registerkarte Schlüsselrotation wird nur auf der Detailseite von KMS-Schlüsseln mit symmetrischer Verschlüsselung angezeigt, deren Schlüsselmaterial AWS KMS generiert wurde (der Ursprung ist AWS_KMS), einschließlich KMS-Schlüsseln mit symmetrischer Verschlüsselung für mehrere Regionen.

   Asymmetrische KMS-Schlüssel, HMAC-KMS-Schlüssel, KMS-Schlüssel mit importiertem Schlüsselmaterial oder KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern können nicht automatisch gedreht werden. Sie können sie jedoch manuell rotieren.
6. Wählen Sie im Abschnitt Automatische Schlüsselrotation die Option Bearbeiten aus.
7. Wählen Sie für Schlüsselrotation die Option Aktivieren aus.

   > **Note**
   >
   > Wenn ein KMS-Schlüssel deaktiviert ist oder noch gelöscht werden AWS KMS muss, wird das Schlüsselmaterial nicht rotiert, und Sie können den Status oder den Rotationszeitraum der automatischen Schlüsselrotation nicht aktualisieren. Aktivieren Sie den KMS-Schlüssel oder brechen Sie den Löschvorgang ab, um die Konfiguration der automatischen Schlüsselrotation zu aktualisieren. Details dazu finden Sie unter So funktioniert die Schlüsselrotation und Wichtige Zustände von AWS KMS Schlüsseln.
8. (Optional) Geben Sie einen Rotationszeitraum zwischen 90 und 2560 Tagen ein. Der Standardwert ist 365 Tage. Wenn Sie keinen benutzerdefinierten Rotationszeitraum angeben, AWS KMS wird das Schlüsselmaterial jedes Jahr rotiert.

   Sie können den RotationPeriodInDays Bedingungsschlüssel kms: verwenden, um die Werte einzuschränken, die Prinzipale für den Rotationsperiode angeben können.
9. Wählen Sie Speichern.

Automatische Schlüsselrotation (AWS KMS API) aktivieren und deaktivieren

Sie können die API AWS Key Management Service (AWS KMS) verwenden, um die automatische Schlüsselrotation zu aktivieren und zu deaktivieren und den aktuellen Rotationsstatus aller vom Kunden verwalteten Schlüssel einzusehen. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Der EnableKeyRotationVorgang ermöglicht die automatische Schlüsselrotation für den angegebenen KMS-Schlüssel. Der DisableKeyRotationVorgang deaktiviert ihn. Um den KMS-Schlüssel bei diesen Operationen zu identifizieren, verwenden Sie seine Schlüssel-ID oder den Schlüssel-ARN. Standardmäßig ist die Schlüsseldrehung für kundenverwaltete KMS-Schlüssel deaktiviert.

Sie können den RotationPeriodInDays Bedingungsschlüssel kms: verwenden, um die Werte einzuschränken, die Prinzipale für den `RotationPeriodInDays` Parameter einer `EnableKeyRotation` Anforderung angeben können.

Das folgende Beispiel aktiviert die Schlüsselrotation mit einer Rotationsperiode von 180 Tagen für den angegebenen KMS-Schlüssel mit symmetrischer Verschlüsselung und verwendet den GetKeyRotationStatusVorgang, um das Ergebnis zu sehen. Anschließend wird die Schlüsselrotation deaktiviert und die Änderung wieder mit GetKeyRotationStatus angezeigt.

```
$ 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"
}

$ 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
}
```

Wie führe ich eine Schlüsselrotation bei Bedarf durch

Sie können eine On-Demand-Rotation des Schlüsselmaterials in vom Kunden verwalteten KMS-Schlüsseln durchführen, unabhängig davon, ob die automatische Schlüsselrotation aktiviert ist oder nicht. Die Deaktivierung der automatischen Rotation (DisableKeyRotation) hat keinen Einfluss auf Ihre Fähigkeit, Rotationen auf Anfrage durchzuführen, und es werden auch keine laufenden On-Demand-Rotationen storniert. Rotationen auf Abruf haben keine Auswirkungen auf bestehende Zeitpläne für automatische Rotationen. Stellen Sie sich zum Beispiel einen KMS-Schlüssel vor, für den die automatische Schlüsselrotation aktiviert ist und für den eine Rotationsperiode von 730 Tagen gilt. Wenn für den Schlüssel eine automatische Rotation am 14. April 2024 geplant ist und Sie am 10. April 2024 eine On-Demand-Rotation durchführen, wird der Schlüssel wie geplant am 14. April 2024 und danach alle 730 Tage automatisch rotiert.

Sie können die Schlüsselrotation bei Bedarf maximal zehnmal pro KMS-Schlüssel durchführen. Sie können die AWS KMS Konsole verwenden, um die Anzahl der verbleibenden On-Demand-Rotationen anzuzeigen, die für einen KMS-Schlüssel verfügbar sind.

Die On-Demand-Schlüsselrotation wird nur für KMS-Schlüssel mit symmetrischer Verschlüsselung unterstützt. Sie können keine On-Demand-Rotation von asymmetrischen KMS-Schlüsseln, HMAC-KMS-Schlüsseln, KMS-Schlüsseln mit importiertem Schlüsselmaterial oder KMS-Schlüsseln in einem benutzerdefinierten Schlüsselspeicher durchführen. Rufen Sie die On-Demand-Rotation für einen Satz verwandter Schlüssel mit mehreren Regionen auf Anforderung für den Primärschlüssel auf.

Autorisierte Benutzer können die AWS KMS Konsole und die AWS KMS API verwenden, um eine On-Demand-Schlüsselrotation einzuleiten und den Status der Schlüsselrotation einzusehen.

Themen

Initiierung der On-Demand-Schlüsselrotation (Konsole)

1. Melden Sie sich bei der AWS Key Management Service (AWS KMS) -Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.

3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel. (Sie können bei Bedarf keine Rotation von durchführen. Von AWS verwaltete Schlüssel Sie werden jedes Jahr automatisch rotiert.)
4. Wählen Sie den Alias oder die Schlüssel-ID eines KMS-Schlüssels.
5. Wählen Sie die Registerkarte Key rotation (Schlüsselrotation).

   Die Registerkarte Schlüsselrotation wird nur auf der Detailseite von KMS-Schlüsseln mit symmetrischer Verschlüsselung angezeigt, deren Schlüsselmaterial AWS KMS generiert wurde (der Ursprung ist AWS_KMS), einschließlich KMS-Schlüsseln mit symmetrischer Verschlüsselung für mehrere Regionen.

   Sie können keine On-Demand-Rotation von asymmetrischen KMS-Schlüsseln, HMAC-KMS-Schlüsseln, KMS-Schlüsseln mit importiertem Schlüsselmaterial oder KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern durchführen. Sie können sie jedoch manuell rotieren.
6. Wählen Sie im Abschnitt Schlüsselrotation auf Anforderung die Option Schlüssel drehen aus.
7. Lesen und berücksichtigen Sie die Warnung und die Informationen über die Anzahl der verbleibenden Drehungen bei Bedarf für den Schlüssel. Wenn Sie entscheiden, dass Sie mit der Rotation auf Anforderung nicht fortfahren möchten, wählen Sie Abbrechen.
8. Wählen Sie die Taste „Drehen“, um die Drehung bei Bedarf zu bestätigen.

> **Note**
>
> Die Rotation nach Bedarf unterliegt letztlich den gleichen Konsistenzeffekten wie andere AWS KMS Verwaltungsvorgänge. Es kann zu einer leichten Verzögerung kommen, bevor das neue Schlüsselmaterial in allen Bereichen von AWS KMS verfügbar ist. Das Banner oben in der Konsole informiert Sie, wenn die On-Demand-Rotation abgeschlossen ist.

Initiierung der On-Demand-Schlüsselrotation (API)AWS KMS

Sie können die API AWS Key Management Service (AWS KMS) verwenden, um eine On-Demand-Schlüsselrotation zu initiieren und den aktuellen Rotationsstatus aller vom Kunden verwalteten Schlüssel einzusehen. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

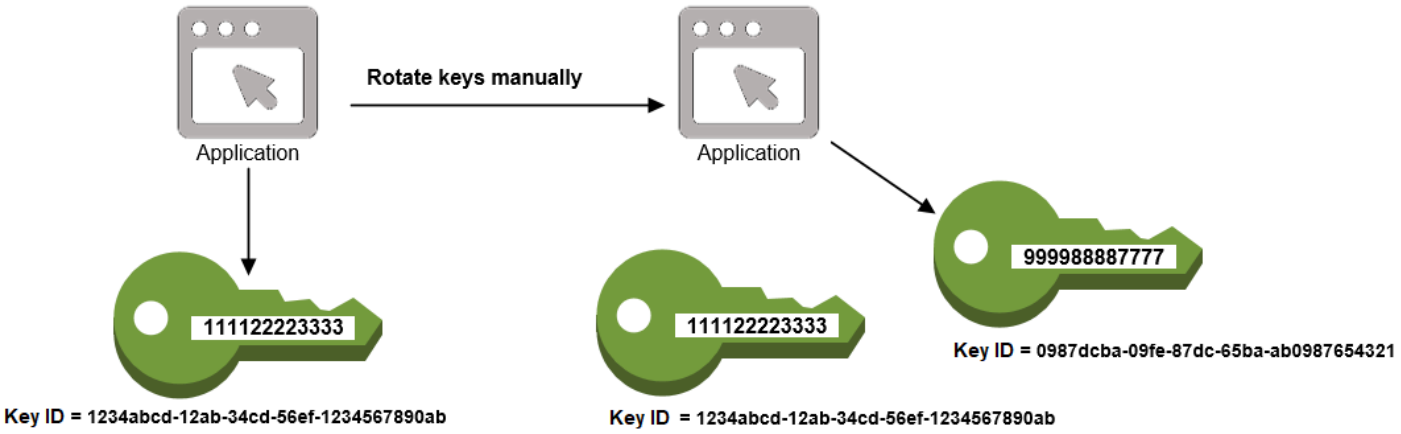
Der RotateKeyOnDemandVorgang initiiert sofort die On-Demand-Schlüsselrotation für den angegebenen KMS-Schlüssel. Um den KMS-Schlüssel bei diesen Operationen zu identifizieren, verwenden Sie seine Schlüssel-ID oder den Schlüssel-ARN.

Im folgenden Beispiel wird eine On-Demand-Schlüsselrotation für den angegebenen KMS-Schlüssel mit symmetrischer Verschlüsselung initiiert und anhand des GetKeyRotationStatusVorgangs überprüft, ob die On-Demand-Rotation ausgeführt wird. `OnDemandRotationStartDate`In der `kms:GetKeyRotationStatus` Antwort werden Datum und Uhrzeit angegeben, zu denen eine laufende On-Demand-Rotation initiiert wurde.

```
$ 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
}
```

Manuelles Rotieren von Schlüsseln

Gegebenenfalls möchten Sie einen neuen KMS-Schlüssel erstellen und diesen anstelle eines aktuellen KMS-Schlüssels verwenden, anstatt die automatische Schlüsseldrehung zu aktivieren. Wenn der neue KMS-Schlüssel andere kryptografische Daten aufweist als der aktuelle KMS-Schlüssel, hat die Verwendung des neuen KMS-Schlüssels den gleichen Effekt wie die Änderung des Schlüsselmaterials in einem vorhandenen KMS-Schlüssel. Das Ersetzen eines KMS-Schlüssels durch einen anderen wird als manuelle Schlüsseldrehung bezeichnet.



Die manuelle Rotation ist eine gute Wahl, wenn Sie KMS-Schlüssel rotieren möchten, die nicht für eine automatische Schlüsselrotation in Frage kommen, z. B. asymmetrische KMS-Schlüssel, HMAC-KMS-Schlüssel, KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern und KMS-Schlüssel mit importiertem Schlüsselmaterial.

> **Note**
>
> Wenn Sie mit der Verwendung des neuen KMS-Schlüssels beginnen, stellen Sie sicher, dass der ursprüngliche KMS-Schlüssel aktiviert bleibt, damit Daten, die mit dem ursprünglichen KMS-Schlüssel verschlüsselt wurden, entschlüsselt werden AWS KMS können.

Wenn Sie KMS-Schlüssel manuell rotieren, müssen Sie auch die Verweise auf die ID oder den ARN des KMS-Schlüssels in Ihren Anwendungen aktualisieren. Dieses Verfahren lässt sich durch Aliasse, die einen Anzeigenamen mit einem KMS-Schlüssel verknüpfen, vereinfachen. Verwenden Sie einen Alias, um auf einen KMS-Schlüssel in Ihren Anwendungen zu verweisen. Wenn Sie dann den von der Anwendung verwendeten KMS-Schlüssel ändern möchten, anstatt den Anwendungscode zu bearbeiten, ändern Sie den Ziel-KMS-Schlüssel des Alias. Details hierzu finden Sie unter Verwenden von Aliassen in Ihren Anwendungen.

> **Note**
>
> Aliase, die auf die neueste Version eines manuell rotierten KMS-Schlüssels verweisen, sind eine gute Lösung für die DescribeKeyOperationen, Verschlüsseln, GenerateDataKeyGenerateDataKeyPairGenerateMac, und Signieren. Aliase sind bei Vorgängen, die KMS-Schlüssel verwalten, nicht zulässig, wie z. B. oder. DisableKeyScheduleKeyDeletion

Wenn Sie den Vorgang Decrypt für manuell rotierte symmetrische KMS-Schlüssel aufrufen, lassen Sie den `KeyId` Parameter im Befehl weg. AWS KMS verwendet automatisch den KMS-Schlüssel, der den Chiffretext verschlüsselt hat.
Der `KeyId` Parameter ist erforderlich, wenn Sie mit einem asymmetrischen KMS-Schlüssel `Decrypt` oder Verify oder VerifyMacmit einem HMAC-KMS-Schlüssel anrufen. Diese Anfragen schlagen fehl, wenn der Wert vom `KeyId`-Parameter ein Alias ist, der nicht mehr auf den KMS-Schlüssel verweist, der die kryptografische Operation ausgeführt hat, z. B. wenn ein Schlüssel manuell gedreht wird. Um diesen Fehler zu vermeiden, müssen Sie den richtigen KMS-Schlüssel für jeden Vorgang verfolgen und angeben.

Um den KMS-Zielschlüssel eines Alias zu ändern, verwenden Sie UpdateAliasOperation in der AWS KMS API. Beispielsweise aktualisiert dieser Befehl den `alias/TestKey`-Alias, so dass er auf einen neuen KMS-Schlüssel verweist. Da der Vorgang keine Ausgabe zurückgibt, zeigt das Beispiel anhand des ListAliasesVorgangs, dass der Alias jetzt einem anderen KMS-Schlüssel zugeordnet ist und das `LastUpdatedDate` Feld aktualisiert wird. Die ListAliases Befehle verwenden den `query`Parameter in AWS CLI , um nur den `alias/TestKey` Alias abzurufen.

```
$ 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
        },
    ]
}
```

Überwachung von AWS KMS keys

Die Überwachung ist ein wichtiger Teil zum Verständnis der Verfügbarkeit, des Zustands und der Nutzung Ihrer AWS KMS keys in AWS KMS und die Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS-Lösungen. Durch das Sammeln von Überwachungsdaten aus allen Teilen Ihrer AWS-Lösung können Sie einen Multipoint-Fehler debuggen, falls ein solcher auftritt. Bevor Sie mit der Überwachung Ihrer KMS-Schlüssel beginnen, erstellen Sie einen Überwachungsplan, der Antworten auf folgende Fragen enthält:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungswerkzeugte verwenden Sie?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer sollte benachrichtigt werden, wenn etwas passiert?

Als nächsten Schritt überwachen Sie Ihre KMS-Schlüssel, um eine Basis für die normale AWS KMS-Nutzung und Erwartungen in Ihrer Umgebung zu erstellen. Während Sie Ihre KMS-Schlüssel überwachen, speichern Sie historische Überwachungsdaten, damit Sie sie mit aktuellen Daten vergleichen können, identifizieren Sie normale Muster und Anomalien und erarbeiten Sie Methoden, um Probleme zu lösen.

Zum Beispiel können Sie AWS KMS-API-Aktivitäten und -Ereignisse überwachen, die Ihre KMS-Schlüssel beeinflussen. Wenn Daten Grenzwerte über- oder unterschreiten, müssen Sie möglicherweise untersuchen oder Korrekturmaßnahmen ergreifen.

Um eine Grundlinie für normale Muster zu erstellen, überwachen Sie folgende Punkte:

- AWS KMS-API-Aktivitäten für Datenebenen-Operationen. Dabei handelt es sich um kryptografische Operationen, die einen KMS-Schlüssel verwenden, z. B. Decrypt , EncryptReEncrypt , und GenerateDataKey.
- AWS KMS-API-Aktivitäten für Steuerebenen-Operationen, die für Sie wichtig sind. Diese Operationen verwalten einen KMS-Schlüssel, und Sie möchten möglicherweise diejenigen überwachen, die die Verfügbarkeit eines KMS-Schlüssels ändern (z. B. ScheduleKeyDeletion, CancelKeyDeletion, DisableKeyEnableKeyImportKeyMaterial, und DeleteImportedKeyMaterial) oder die Zugriffskontrolle eines KMS-Schlüssels ändern (z. B. PutKeyPolicy und RevokeGrant).
- Andere AWS KMS-Metriken (wie die noch verbleibende Zeit, bis Ihr importiertes Schlüsselmaterial abläuft, und Ereignisse (z. B. das Ablaufen des importierten Schlüsselmaterials oder das Löschen oder eine Schlüsseldrehung eines KMS-Schlüssels).

Überwachungstools

AWS bietet verschiedene Werkzeuge, mit denen Sie Ihre KMS-Schlüssel überwachen können. Sie können einige dieser Tools so konfigurieren, dass diese die Überwachung für Sie übernehmen, während bei anderen Tools ein manuelles Eingreifen nötig ist. Wir empfehlen, dass Sie die Überwachungsaufgaben möglichst automatisieren.

Automatisierte Überwachungstools

Sie können die folgenden automatisierten Überwachungstools verwenden, um Ihre KMS-Schlüssel zu verfolgen und zu berichten, wenn sich etwas geändert hat.

- AWS CloudTrail Protokollüberwachung – Teilen Sie Protokolldateien zwischen Konten, überwachen Sie CloudTrail Protokolldateien in Echtzeit, indem Sie sie an - CloudWatch Protokolle senden, schreiben Sie Anwendungen zur Protokollverarbeitung mit der CloudTrail Processing Library und überprüfen Sie, ob sich Ihre Protokolldateien nach der Bereitstellung durch nicht geändert haben CloudTrail. Weitere Informationen finden Sie unter Arbeiten mit CloudTrail Protokolldateien im AWS CloudTrail -Benutzerhandbuch.
- Amazon CloudWatch -Alarme – Überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führen Sie eine oder mehrere Aktionen aus, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über eine Reihe von Zeiträumen basieren. Die Aktion ist eine Benachrichtigung, die an ein Amazon Simple Notification Service (Amazon SNS)-Thema oder eine Amazon EC2 Auto Scaling-Richtlinie gesendet wird. - CloudWatch Alarme rufen keine Aktionen auf, nur weil sie sich in einem bestimmten Status

befinden. Der Status muss geändert und für eine bestimmte Anzahl von Zeiträumen beibehalten worden sein. Weitere Informationen finden Sie unter Überwachung mit Amazon CloudWatch.

- Amazon EventBridge – Ordnen Sie Ereignisse zu und leiten Sie sie an eine oder mehrere Zielfunktionen oder Streams weiter, um Statusinformationen zu erfassen und bei Bedarf Änderungen vorzunehmen oder Korrekturmaßnahmen zu ergreifen. Weitere Informationen finden Sie unter Überwachung mit Amazon EventBridge und im Amazon- EventBridge Benutzerhandbuch.
- Amazon CloudWatch Logs – Überwachen, Speichern und Zugriff auf Ihre Protokolldateien von AWS CloudTrail oder anderen Quellen. Weitere Informationen finden Sie im Amazon- CloudWatch Logs-Benutzerhandbuch.

Manuelle Überwachungstools

Ein weiterer wichtiger Bestandteil der Überwachung von KMS-Schlüsseln ist die manuelle Überwachung derjenigen Elemente, die die CloudWatch Alarme und Ereignisse nicht abdecken. Die AWS Dashboards AWS KMS, CloudWatchAWS Trusted Advisor, und andere bieten einen at-a-glance Überblick über den Zustand Ihrer AWS Umgebung.

Sie können die Seiten Von AWS verwaltete Schlüssel und Customer managed keys (kundenverwaltete Schlüssel) der AWS KMS-Konsole anpassen, um folgende Informationen zu jedem KMS-Schlüssel anzuzeigen:

- Schlüssel-ID
- Status
- Erstellungsdatum
- Ablaufdatum (für KMS-Schlüssel mit importiertem Schlüsselmaterial)
- Ursprung
- Benutzerdefinierte Schlüsselspeicher-ID (für KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern)

Das CloudWatch -Konsolen-Dashboard zeigt Folgendes an:

- Aktuelle Alarme und Status
- Diagramme mit Alarmen und Ressourcen
- Servicestatus

Darüber hinaus können Sie mit Folgendes CloudWatch tun:

- Erstellen angepasster Dashboards zur Überwachung der gewünschten Services.
- Aufzeichnen von Metrikdaten, um Probleme zu beheben und Trends zu erkennen
- Durchsuchen und Suchen aller AWS-Ressourcenmetriken
- Erstellen und Bearbeiten von Alarmen, um über Probleme benachrichtigt zu werden

AWS Trusted Advisor kann Ihnen helfen, Ihre AWS Ressourcen zu überwachen, um Leistung, Zuverlässigkeit, Sicherheit und Kosteneffektivität zu verbessern. Vier Trusted Advisor-Prüfungen stehen allen Benutzern zur Verfügung; mehr als 50 Überprüfungen stehen Benutzern mit einem Business- oder Enterprise-Supportplan zur Verfügung. Weitere Informationen finden Sie unter AWS Trusted Advisor.

AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail

AWS KMS ist in einen Dienst integriert AWS CloudTrail, der alle Aufrufe AWS KMS von Benutzern, Rollen und anderen AWS Diensten aufzeichnet. CloudTrail erfasst alle API-Aufrufe AWS KMS als Ereignisse, einschließlich Aufrufe von der AWS KMS Konsole, AWS KMS APIs, AWS CloudFormation Vorlagen, AWS Command Line Interface (AWS CLI) und AWS Tools for PowerShell.

CloudTrail protokolliert alle AWS KMS Operationen, einschließlich schreibgeschützter Operationen wie ListAliasesund GetKeyRotationStatus, Operationen zur Verwaltung von KMS-Schlüsseln wie CreateKeyund und PutKeyPolicykryptografischer Operationen wie GenerateDataKeyund Decrypt. Außerdem werden interne Vorgänge protokolliert, die für AWS KMS Sie erforderlich sind, z. B., DeleteExpiredKeyMaterial, und DeleteKey. SynchronizeMultiRegionKeyRotateKey

CloudTrail protokolliert erfolgreiche Vorgänge und fehlgeschlagene Aufrufversuche, z. B. wenn dem Anrufer der Zugriff auf eine Ressource verweigert wird. Kontoübergreifende Vorgänge auf KMS-Schlüssel in anderen Konten werden sowohl im Konto des Anrufers als auch im Konto des KMS-Schlüsselbesitzers protokolliert. Kontoübergreifende AWS KMS Anfragen, die abgelehnt werden, weil der Zugriff verweigert wurde, werden jedoch nur im Konto des Anrufers protokolliert.

Aus Sicherheitsgründen werden einige Felder in den AWS KMS Protokolleinträgen weggelassen, z. B. der `Plaintext` Parameter einer Verschlüsselungsanforderung und die Antwort auf GetKeyPolicyoder andere kryptografische Operationen. Um die Suche nach CloudTrail Protokolleinträgen für bestimmte KMS-Schlüssel zu vereinfachen, AWS KMS fügt der Schlüssel-ARN des betroffenen KMS-Schlüssels dem `responseElements` Feld in den Protokolleinträgen für einige

AWS KMS Schlüsselverwaltungsvorgänge hinzu, auch wenn der API-Vorgang den Schlüssel-ARN nicht zurückgibt.

Standardmäßig werden zwar alle AWS KMS Aktionen als CloudTrail Ereignisse protokolliert, Sie können jedoch AWS KMS Aktionen von einem CloudTrail Trail ausschließen. Details hierzu finden Sie unter AWS KMS Ereignisse aus einer Spur ausschließen.

Weitere Informationen:

- CloudTrail Protokollbeispiele für AWS KMS Operationen in einer AWS Nitro-Enklave finden Sie unter. Überwachung von Anfragen für Nitro-Enklaven

Themen

Ereignisse protokollieren CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet AWS KMS, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können in Ihrem AWS-Konto die neusten Ereignisse anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für AWS KMS, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere konfigurieren, AWS-Services um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen

- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch. Weitere Informationen zu anderen Methoden zur Überwachung der Nutzung Ihrer KMS-Schlüssel finden Sie unter Überwachung von AWS KMS keys.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root-Benutzeranmeldeinformationen oder IAM-Benutzeranmeldeinformationen gestellt wurde.
- Wurde die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer ausgeführt?
- Wenn die Anfrage von einem anderen gestellt wurde AWS-Service.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Suchen nach Ereignissen in CloudTrail

Verwenden Sie die CloudTrail Konsole oder den CloudTrail LookupEventsVorgang, um nach CloudTrail Protokolleinträgen zu suchen. CloudTrail unterstützt zahlreiche Attributwerte zum Filtern Ihrer Suche, einschließlich Ereignisname, Benutzername und Ereignisquelle.

Füllt die folgenden AWS KMS Protokolleintragsfelder aus CloudTrail, AWS KMS um Ihnen die Suche nach CloudTrail Protokolleinträgen zu erleichtern.

> **Note**
>
> Füllt ab Dezember 2022 die Attribute Ressourcentyp und Ressourcenname in allen Verwaltungsvorgängen aus, die einen bestimmten KMS-Schlüssel ändern. AWS KMS Diese Attributwerte können in älteren CloudTrail Einträgen für die folgenden Operationen Null sein: CreateAliasCreateGrantDeleteAlias, DeleteImportedKeyMaterial, ImportKeyMaterial, ReplicateKey, RetireGrant, RevokeGrantUpdateAlias, und UpdatePrimaryRegion.

Attribut	Wert	Protokolleinträge
Ereignisquelle (`EventSource`)	`kms.amazonaws.com`	Alle Vorgänge.
Ressourcentyp (`ResourceType`)	`AWS::KMS::Key`	Verwaltungsvorgänge, die einen bestimmten KMS-Schlüssel ändern, z. B. `CreateKey` und `EnableKey`, aber nicht `ListKeys`.
Ressourcenname (`ResourceName`)	Schlüssel-ARN (oder Schlüssel-ID und Schlüssel-ARN)	Verwaltungsvorgänge, die einen bestimmten KMS-Schlüssel ändern, z. B. `CreateKey` und `EnableKey`, aber nicht `ListKeys`.

Um Ihnen das Auffinden von Protokolleinträgen für Verwaltungsvorgänge mit bestimmten KMS-Schlüsseln zu erleichtern, AWS KMS zeichnet es den Schlüssel-ARN des betroffenen KMS-Schlüssels im `responseElements.keyId` Element des Protokolleintrags auf, auch wenn der AWS KMS API-Vorgang den Schlüssel-ARN nicht zurückgibt.

Beispielsweise gibt ein erfolgreicher Aufruf des DisableKeyVorgangs keine Werte in der Antwort zurück, aber statt eines Nullwerts enthält der `responseElements.keyId` Wert im DisableKey Protokolleintrag den Schlüssel ARN des deaktivierten KMS-Schlüssels.

Diese Funktion wurde im Dezember 2022 hinzugefügt und wirkt sich auf die folgenden CloudTrail Protokolleinträge aus: CreateAliasCreateGrantDeleteAlias, DeleteKey, DisableKey, EnableKey, EnableKeyRotation, ImportKeyMaterial, RotateKey, SynchronizeMultiRegionKey, TagResource, UntagResource, UpdateAlias, und UpdatePrimaryRegion.

AWS KMS Ereignisse aus einer Spur ausschließen

Um die Nutzung und Verwaltung ihrer AWS KMS Ressourcen aufzuzeichnen, verlassen sich die meisten AWS KMS Benutzer auf die Ereignisse in einem CloudTrail Trail. Der Trail kann eine wertvolle Datenquelle für die Überwachung kritischer Ereignisse sein, wie z. B. das Erstellen, Deaktivieren und Löschen AWS KMS keys, Ändern von Schlüsselrichtlinien und die Verwendung

Ihrer KMS-Schlüssel durch AWS Dienste in Ihrem Namen. In einigen Fällen können Ihnen die Metadaten in einem CloudTrail Protokolleintrag, z. B. der Verschlüsselungskontext bei einem Verschlüsselungsvorgang, dabei helfen, Fehler zu vermeiden oder zu beheben.

Da jedoch eine große Anzahl von Ereignissen generiert werden AWS KMS kann, AWS CloudTrail können Sie AWS KMS Ereignisse aus einer Spur ausschließen. Diese Einstellung pro Trail schließt alle AWS KMS Ereignisse aus. Sie können bestimmte AWS KMS Ereignisse nicht ausschließen.

> **Warning**
>
> Durch das Ausschließen von AWS KMS Ereignissen aus einem CloudTrail Protokoll können Aktionen, die Ihre KMS-Schlüssel verwenden, verschleiert werden. Seien Sie vorsichtig, wenn Sie Prinzipalen die `cloudtrail:PutEventSelectors`-Berechtigung erteilen, die zum Ausführen dieser Operation erforderlich ist.

So schließen Sie AWS KMS Ereignisse aus einem Trail aus:

- Verwenden Sie in der CloudTrail Konsole die Einstellung Ereignisse des Schlüsselverwaltungsdienstes protokollieren, wenn Sie einen Trail erstellen oder einen Trail aktualisieren. Anweisungen finden Sie AWS Management Console im AWS CloudTrail Benutzerhandbuch unter Protokollieren von Management-Ereignissen mit dem.
- Verwenden Sie in der CloudTrail API den PutEventSelectorsVorgang. Fügen Sie das Attribut `ExcludeManagementEventSources` mit dem Wert `kms.amazonaws.com` zu Ihren Ereignisselektoren hinzu. Ein Beispiel finden Sie im AWS CloudTrail Benutzerhandbuch unter Beispiel: Ein Trail, der keine AWS Key Management Service Ereignisse protokolliert.

Sie können diesen Ausschluss jederzeit deaktivieren, indem Sie die Konsoleneinstellung oder die Ereignisselektoren für einen Trail ändern. Der Trail beginnt dann mit der Aufzeichnung von AWS KMS Ereignissen. AWS KMS Ereignisse, die während der Gültigkeit des Ausschlusses eingetreten sind, können jedoch nicht wiederhergestellt werden.

Wenn Sie AWS KMS Ereignisse mithilfe der Konsole oder der API ausschließen, wird der daraus resultierende CloudTrail `PutEventSelectors` API-Vorgang auch in Ihren CloudTrail Protokollen protokolliert. Wenn AWS KMS Ereignisse nicht in Ihren CloudTrail Protokollen erscheinen, suchen Sie nach einem `PutEventSelectors` Ereignis, bei dem das `ExcludeManagementEventSources` Attribut auf gesetzt ist`kms.amazonaws.com`.

Beispiele für AWS KMS Protokolleinträge

AWS KMS schreibt Einträge in Ihr CloudTrail Protokoll, wenn Sie einen AWS KMS Vorgang aufrufen und wenn ein AWS Dienst einen Vorgang in Ihrem Namen aufruft. AWS KMS schreibt auch einen Eintrag, wenn er eine Operation für Sie aufruft. Zum Beispiel schreibt es einen Eintrag, wenn es einen KMS-Schlüssel löscht, den Sie zum Löschen vorgesehen haben.

Die folgenden Themen enthalten Beispiele für CloudTrail Protokolleinträge für AWS KMS Operationen.

Beispiele für CloudTrail Protokolleinträge von Anfragen an AWS KMS von AWS Nitro Enclaves finden Sie unter. Überwachung von Anfragen für Nitro-Enklaven

Themen

- 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

- Verify
- Amazon EC2 – Beispiel 1
- Amazon EC2 – Beispiel 2

CancelKeyDeletion

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der CancelKeyDeletion-Operation generiert wird. Informationen zum Löschen von AWS KMS keys finden Sie unter Löschen von 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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der `ConnectCustomKeyStore`-Operation generiert wird. Hinweise zum Verbinden von benutzerdefinierten Schlüsselspeichern finden Sie unter Herstellen und Trennen der Verbindung eines AWS CloudHSM-Schlüsselspeichers.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -CreateAliasOperation. Das `resources`-Element enthält Felder für den Alias und KMS-Schlüsselressourcen. Weitere Informationen zum Erstellen von Aliassen in AWS KMS finden Sie unter Erstellen eines Alias.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen des `CreateCustomKeyStore`-Vorgangs an einem AWS CloudHSM-Schlüsselspeicher generiert wird. Hinweise zum Erstellen von benutzerdefinierten Schlüsselspeichern finden Sie unter Einen AWS CloudHSM-Schlüsselspeicher erstellen.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -CreateGrantOperation. Weitere Informationen zum Erstellen von Erteilungen in AWS KMS finden Sie unter Erteilungen in AWS KMS.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

```
{
  "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

Diese Beispiele zeigen AWS CloudTrail Protokolleinträge für die -CreateKeyOperation.

Ein `CreateKey` Protokolleintrag kann sich aus einer `CreateKey` Anforderung oder der `CreateKey` Operation für eine ReplicateKey Anforderung ergeben.

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag für eine -CreateKeyOperation, die einen KMS-Schlüssel mit symmetrischer Verschlüsselung erstellt. Weitere Informationen zum Erstellen und Verwalten von KMS-Schlüsseln finden Sie unter Erste Schritte.

```
{
    "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"
}
```

Das folgende Beispiel zeigt das CloudTrail Protokoll einer -`CreateKey`Operation, die einen KMS-Schlüssel mit symmetrischer Verschlüsselung in einem AWS CloudHSM -Schlüsselspeicher erstellt.

```
{
    "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": "{\"keyHandle\":\"19\",\"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"
}
```


Das folgende Beispiel zeigt das CloudTrail Protokoll einer -`CreateKey`Operation, die einen KMS-Schlüssel mit symmetrischer Verschlüsselung in einem externen Schlüsselspeicher erstellt.


```
{
    "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

Diese Beispiele zeigen AWS CloudTrail-Protokolleinträge für die Decrypt-Operation.

Der CloudTrail Protokolleintrag für eine -`Decrypt`Operation enthält immer den `encryptionAlgorithm` in der , `requestParameters` auch wenn der Verschlüsselungsalgorithmus nicht in der Anforderung angegeben wurde. Der Chiffretext in der Anforderung und der Klartext in der Antwort werden weggelassen.

Themen

- Entschlüsselung mit einem symmetrischen Standardverschlüsselungsschlüssel
- Fehler bei der Entschlüsselung mit einem symmetrischen Standardverschlüsselungsschlüssel
- Entschlüsselung mit einem KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher
- Entschlüsselung mit einem KMS-Schlüssel in einem externen Schlüsselspeicher
- Fehler bei der Entschlüsselung mit einem KMS-Schlüssel in einem externen Schlüsselspeicher

Entschlüsselung mit einem symmetrischen Standardverschlüsselungsschlüssel

Im Folgenden finden Sie ein Beispiel für einen CloudTrail Protokolleintrag für eine -`Decrypt`Operation mit einem symmetrischen Standardverschlüsselungsschlüssel.


```
{
    "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"
```

```
}
```

Fehler bei der Entschlüsselung mit einem symmetrischen Standardverschlüsselungsschlüssel

Der folgende CloudTrail Beispielprotokolleintrag zeichnet einen fehlgeschlagenen `Decrypt` Vorgang mit einem standardmäßigen KMS-Schlüssel mit symmetrischer Verschlüsselung auf. Die Ausnahme (`errorCode`) und die Fehlermeldung (`errorMessage`) sind enthalten und helfen Ihnen, den Fehler zu beheben.

In diesem Fall war der in der `Decrypt`-Anfrage angegebene symmetrische KMS-Verschlüsselungsschlüssel nicht der symmetrische KMS-Verschlüsselungsschlüssel, der zum Verschlüsseln der Daten verwendet wurde.

```
{
    "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"
}
```

Entschlüsselung mit einem KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher

Der folgende CloudTrail Beispielprotokolleintrag zeichnet eine -`Decrypt`Operation mit einem KMS-Schlüssel in einem AWS CloudHSM -Schlüsselspeicher auf. Alle Protokolleinträge für kryptografische Operationen mit einem KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher umfassen ein `additionalEventData`-Feld mit der `customKeyStoreId`. Die `additionalEventData` ist nicht in der Anfrage angegeben.

```
{
    "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"
}
```

Entschlüsselung mit einem KMS-Schlüssel in einem externen Schlüsselspeicher

Der folgende CloudTrail Beispielprotokolleintrag zeichnet eine -`Decrypt`Operation mit einem KMS-Schlüssel in einem externen Schlüsselspeicher auf. Zusätzlich zur `customKeyStoreId` enthält das `additionalEventData`-Feld die externe Schlüssel-ID (`XksKeyId`). Die `additionalEventData` ist nicht in der Anfrage angegeben.

```
{
    "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"
}
```

Fehler bei der Entschlüsselung mit einem KMS-Schlüssel in einem externen Schlüsselspeicher

Der folgende CloudTrail Beispielprotokolleintrag zeichnet zusätzlich CloudWatch zu erfolgreichen Anforderungen eine fehlgeschlagene Anforderung für eine `Decrypt` Operation mit einem KMS-Schlüssel in einem externen Schlüsselspeicher auf. Bei der Aufzeichnung eines Fehlers enthält der CloudTrail Protokolleintrag die Ausnahme (errorCode) und die zugehörige Fehlermeldung (errorMessage).

Wenn die fehlgeschlagene Anforderung Ihren externen Schlüsselspeicher-Proxy erreicht hat, wie in diesem Beispiel, können Sie den `requestId`-Wert verwenden, um die fehlgeschlagene Anforderung einer entsprechenden Anfrage zuzuordnen, die Ihr externer Schlüsselspeicher-Proxy protokolliert, sofern Ihr Proxy sie bereitstellt.

Hilfe zu `Decrypt`-Anfragen in externen Schlüsselspeichern finden Sie unter Entschlüsselungsfehler.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -DeleteAliasOperation. Weitere Informationen zum Löschen von Aliassen finden Sie unter Löschen eines Alias.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der `DeleteCustomKeyStore`-Operation generiert wird. Hinweise zum Erstellen von benutzerdefinierten Schlüsselspeichern finden Sie unter Löschen eines AWS CloudHSM-Schlüsselspeichers.

```
{
    "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

Wenn Sie Schlüsselmaterial in einen AWS KMS key (KMS-Schlüssel) importieren, können Sie ein Ablaufdatum und eine Ablaufzeit für dieses Schlüsselmaterial festlegen. AWS KMS zeichnet einen Eintrag in Ihrem CloudTrail Protokoll auf, wenn Sie das Schlüsselmaterial importieren (mit den Ablaufeinstellungen) und wenn das abgelaufene Schlüsselmaterial AWS KMSlöscht. Weitere Informationen zum Erstellen eines KMS-Schlüssels mit importiertem Schlüsselmaterial finden Sie unter Schlüsselmaterial für AWS KMS Schlüssel importieren.

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der generiert wird, wenn AWS KMS das abgelaufene Schlüsselmaterial löscht.


```
{
    "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

Wenn Sie Schlüsselmaterial in einen KMS-Schlüssel importieren, können Sie das importierte Schlüsselmaterial jederzeit mithilfe der -DeleteImportedKeyMaterialOperation löschen. Wenn Sie importiertes Schlüsselmaterial von einem KMS-Schlüssel löschen, ändert sich der Schlüsselstatus des KMS-Schlüssels auf `PendingImport`, und der KMS-Schlüssel kann nicht in kryptografischen Vorgängen verwendet werden. Details hierzu finden Sie unter Löschen von importiertem Schlüsselmaterial.

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag für die `DeleteImportedKeyMaterial`-Operation.

```
{
    "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

Diese Beispiele zeigen den AWS CloudTrail-Protokolleintrag, der generiert wird, wenn ein KMS-Schlüssel gelöscht wird. Um einen KMS-Schlüssel zu löschen, verwenden Sie die -ScheduleKeyDeletionOperation. Nach Ablauf der angegebenen Wartezeit AWS KMSlöscht den KMS-Schlüssel und zeichnet einen Eintrag wie den folgenden in Ihrem CloudTrail Protokoll auf, um dieses Ereignis aufzuzeichnen.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

Ein Beispiel für den CloudTrail Protokolleintrag für die -`ScheduleKeyDeletion`Operation finden Sie unter ScheduleKeyDeletion. Weitere Informationen zum Erstellen und Verwalten von KMS-Schlüsseln finden Sie unter Erste Schritte.

Der folgende CloudTrail Beispielprotokolleintrag zeichnet eine -`DeleteKey`Operation eines KMS-Schlüssels mit Schlüsselmaterial in aufAWS 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"
}
```

Der folgende CloudTrail Protokolleintrag zeichnet einen `DeleteKey` Vorgang eines KMS-Schlüssels in einem AWS CloudHSM benutzerdefinierten Schlüsselspeicher auf.

```
{
    "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": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"01\",\"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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der DescribeCustomKeyStores-Operation generiert wird. Hinweise zum Anzeigen von

benutzerdefinierten Schlüsselspeichern finden Sie unter Anzeigen eines AWS CloudHSM-Schlüsselspeichers.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -DescribeKeyOperation. AWS KMS zeichnet einen Eintrag wie den folgenden auf, wenn Sie die -`DescribeKey`Operation aufrufen oder KMS-Schlüssel in der -Konsole anzeigen. AWS KMS Dieser Aufruf ist das Ergebnis der Anzeige eines Schlüssels in der AWS KMS-Managementkonsole.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -DisableKeyOperation. Informationen zum Aktivieren und Deaktivieren von AWS KMS keys in AWS KMS finden Sie unterAktivieren und Deaktivieren von Schlüsseln.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.


```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der DisableKeyRotation-Operation generiert wird. Weitere Informationen zur automatischen Schlüsselrotation finden Sie unter Rotierend 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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der `DisconnectCustomKeyStore`-Operation generiert wird. Hinweise zum Trennen von benutzerdefinierten Schlüsselspeichern finden Sie unter Herstellen und Trennen der Verbindung eines AWS CloudHSM-Schlüsselspeichers.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -EnableKeyOperation. Informationen zum Aktivieren und Deaktivieren von AWS KMS keys in AWS KMS finden Sie unterAktivieren und Deaktivieren von Schlüsseln.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.


```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail Protokolleintrag eines Aufrufs der EnableKeyRotationOperation. Ein Beispiel für den CloudTrail Protokolleintrag, der geschrieben wird, wenn der Schlüssel gedreht wird, finden Sie unterRotateKey. Weitere Informationen zur Drehung von AWS KMS keys finden Sie unter Rotierend AWS KMS keys.

> **Note**
>
> Das rotation-period ist ein optionaler Anforderungsparameter. Wenn Sie bei der Aktivierung der automatischen Schlüsselrotation keinen Rotationszeitraum angeben, ist der Standardwert 365 Tage.

CloudTrail Protokolleinträge für diesen Vorgang, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `responseElements.keyId` Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.


```
{
    "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"
}
```

Encrypt

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag für die Encrypt-Operation.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateDataKeyOperation.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateDataKeyPairOperation. In diesem Beispiel wird eine Operation aufgezeichnet, die ein RSA-Schlüsselpaar generiert, das mit einer symmetrischen Verschlüsselung AWS KMS key verschlüsselt ist.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateDataKeyPairWithoutPlaintextOperation. In diesem Beispiel wird eine Operation aufgezeichnet, die ein RSA-Schlüsselpaar generiert, das mit einer symmetrischen Verschlüsselung AWS KMS key verschlüsselt ist.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateDataKeyWithoutPlaintextOperation.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateMacOperation.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GenerateRandomOperation. Da diese Operation keinen AWS KMS key nutzt, ist das `resources`-Feld leer.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -GetKeyPolicyOperation. Informationen zum Anzeigen der Schlüsselrichtlinie für einen KMS-Schlüssel finden Sie unter Anzeigen einer Schlüsselrichtlinie.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail Protokolleintrag für den GetKeyRotationStatusVorgang. Informationen zur automatischen und bedarfsgesteuerten Rotation von Schlüsselmaterial für einen KMS-Schlüssel finden Sie unterRotierend 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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag, der generiert wird, wenn Sie die -GetParametersForImportOperation verwenden. Dieser Vorgang gibt den öffentlichen Schlüssel und Import-Token zurück, den Sie beim Importieren von Schlüsselmaterial in einen

KMS-Schlüssel verwenden. Derselbe CloudTrail Eintrag wird aufgezeichnet, wenn Sie die -`GetParametersForImport`Operation oder die AWS KMS Konsole verwenden, um den öffentlichen Schlüssel und das Import-Token herunterzuladen.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag, der generiert wird, wenn Sie die -ImportKeyMaterialOperation verwenden. Derselbe CloudTrail Eintrag wird aufgezeichnet, wenn Sie die -`ImportKeyMaterial`Operation oder die -AWS KMSKonsole verwenden, um Schlüsselmaterial in ein zu importierenAWS KMS key.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im -`responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -ListAliasesOperation. Da diese Operation keinen bestimmten Alias oder AWS KMS key nutzt, ist das `resources`-Feld leer. Weitere Informationen zum Anzeigen von Aliassen in AWS KMS finden Sie unter Anzeigen von Aliassen.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -ListGrantOperation. Weitere Informationen über Erteilungen in AWS KMS finden Sie unter Erteilungen 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

Das folgende Beispiel zeigt einen AWS CloudTrail Protokolleintrag für den ListKeyRotationsVorgang. Informationen zur automatischen und bedarfsgesteuerten Rotation von Schlüsselmaterial für einen KMS-Schlüssel finden Sie unterRotierend 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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der PutKeyPolicy-Operation generiert wird. Informationen zur Aktualisierung einer Schlüsselrichtlinie finden Sie unter Ändern einer Schlüsselrichtlinie.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -ReEncryptOperation. Das `resources`-Feld in diesem Protokolleintrag gibt zwei AWS KMS keys an, den Quell-KMS-Schlüssel und den Ziel-KMS-Schlüssel in dieser Reihenfolge.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der ReplicateKey-Operation generiert wird. Eine -`ReplicateKey`Anforderung führt zu einer -`ReplicateKey`Operation und einer -CreateKeyOperation.

Informationen zum Replizieren von multiregionalen Schlüsseln finden Sie unter Erstellen von multiregionalen Replikatschlüsseln.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der RetireGrant-Operation generiert wird. Informationen zur Außerbetriebnahme von Erteilungen finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.


```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der RevokeGrant-Operation generiert wird. Informationen zum Widerrufen von Erteilungen finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.

```
{
    "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

Diese Beispiele zeigen die AWS CloudTrail Protokolleinträge für die rotierenden Operationen AWS KMS keys. Weitere Informationen zur Drehung von KMS-Schlüssel finden Sie unter Rotierend AWS KMS keys.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für den Vorgang, bei dem ein KMS-Schlüssel mit symmetrischer Verschlüsselung rotiert wird, für den die automatische Schlüsselrotation aktiviert ist. Hinweise zur Aktivierung der automatischen Rotation finden Sie unter. So aktivieren und deaktivieren Sie die automatische Schlüsselrotation:

Ein Beispiel für den CloudTrail Protokolleintrag, der den `EnableKeyRotation` Vorgang aufzeichnet, finden Sie unterEnableKeyRotation.

```
{
    "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"
}
```


Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für einen RotateKeyOnDemandVorgang. Hinweise zur bedarfsgesteuerten Rotation von KMS-Schlüsseln mit symmetrischer Verschlüsselung finden Sie unterWie führe ich eine Schlüsselrotation bei Bedarf durch.

Ein Beispiel für den CloudTrail Protokolleintrag, der den `RotateKeyOnDemand` Vorgang aufzeichnet, finden Sie unterRotateKeyOnDemand.


```
{
```

```
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail Protokolleintrag für den RotateKeyOnDemandVorgang. Ein Beispiel für den CloudTrail Protokolleintrag, der geschrieben wird, wenn der Schlüssel gedreht wird, finden Sie unterRotateKey. Weitere Informationen zur On-Demand-Rotation von Schlüsselmaterial für einen KMS-Schlüssel finden Sie unterWie führe ich eine Schlüsselrotation bei Bedarf durch.

```
{
    "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

Diese Beispiele zeigen AWS CloudTrail Protokolleinträge für die -ScheduleKeyDeletionOperation.

Ein Beispiel für den CloudTrail Protokolleintrag, der beim Löschen des Schlüssels geschrieben wird, finden Sie unter DeleteKey. Informationen zum Löschen von AWS KMS keys finden Sie unter Löschen von AWS KMS keys.

Im folgenden Beispiel wird eine `ScheduleKeyDeletion`-Anforderung für einen einzelregionalen KMS-Schlüssel aufgezeichnet.


```
{
    "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"
}
```

Im folgenden Beispiel wird eine `ScheduleKeyDeletion`-Anforderung für einen multiregionalen KMS-Schlüssel mit Replikatschlüsseln aufgezeichnet.

AWS KMS löscht einen multiregionalen Schlüssel erst dann, wenn alle seine Replikatschlüssel gelöscht wurden. Daher ist im `responseElements`-Feld der `keyState` `PendingReplicaDeletion` und das `deletionDate`-Feld wird ausgelassen.

```
{
    "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"
}
```

Im folgenden Beispiel wird eine `ScheduleKeyDeletion`-Anforderung für einen KMS-Schlüssel in einem benutzerdefinierten AWS CloudHSM-Schlüsselspeicher aufgezeichnet.

```
{
    "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": "[{\"keyHandle\":\"01\",\"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

Diese Beispiele zeigen AWS CloudTrail-Protokolleinträge für die Sign-Operation.

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag für eine Sign-Operation, die einen asymmetrischen RSA-KMS-Schlüssel verwendet, um eine digitale Signatur für eine Datei zu generieren.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der generiert wird, wenn AWS KMS einen multiregionalen Schlüssel synchronisiert. Beim Synchronisieren werden regionsübergreifende Aufrufe zum Kopieren der gemeinsamen Eigenschaften eines multiregionalen Primärschlüssels zu seinen Replikatschlüsseln verwendet. AWS KMS synchronisiert multiregionale Schlüssel in regelmäßigen Abständen, um sicherzustellen, dass alle zugehörige multiregionale Schlüssel das gleiche Schlüsselmaterial haben.

Das -`resources`Element des CloudTrail Protokolleintrags enthält den Schlüssel-ARN des multiregionalen Primärschlüssels, einschließlich seines AWS-Region. Die zugehörigen multiregionalen Replikatschlüssel und ihre Regionen werden in diesem Protokolleintrag nicht aufgeführt.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im -`responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag eines Aufrufs der -TagResourceOperation zum Hinzufügen eines Tags mit einem Tag-Schlüssel von `Department` und einem Tag-Wert von `IT`.

Ein Beispiel für einen -`UntagResource` CloudTrail Protokolleintrag, der beim Rotieren des Schlüssels geschrieben wird, finden Sie unter UntagResource. Weitere Informationen über das Markieren von AWS KMS keys finden Sie unter Tagging von Schlüsseln.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag eines Aufrufs der -UntagResourceOperation zum Löschen eines Tags mit dem Tag-Schlüssel `Dept`.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

Ein Beispiel für einen -`TagResource` CloudTrail Protokolleintrag finden Sie unter TagResource. Weitere Informationen über das Markieren von AWS KMS keys finden Sie unter Tagging von Schlüsseln.

```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -UpdateAliasOperation. Das `resources`-Element enthält Felder für den Alias und KMS-Schlüsselressourcen. Weitere Informationen zum Erstellen von Aliassen in AWS KMS finden Sie unter Erstellen eines Alias.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im -`responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der generiert wurde, als die `UpdateCustomKeyStore`-Operation aufgerufen wurde, um die Cluster-ID für einen benutzerdefinierten Schlüsselspeicher zu aktualisieren. Hinweise zum Bearbeiten von benutzerdefinierten Schlüsselspeichern finden Sie unter Einstellungen des AWS CloudHSM-Schlüsselspeichers bearbeiten.

```
{
    "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

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag, der durch Aufrufen der UpdateKeyDescription-Operation generiert wird.

```
{
    "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

Das folgende Beispiel zeigt die AWS CloudTrail Protokolleinträge, die durch Aufrufen der -UpdatePrimaryRegionOperation für einen multiregionalen Schlüssel generiert werden. ???

Die `UpdatePrimaryRegion` Operation schreibt zwei CloudTrail Protokolleinträge: einen in der Region mit dem multiregionalen Primärschlüssel, der in einen Replikatschlüssel konvertiert wird, und einen in der Region mit einem multiregionalen Replikatschlüssel, der in einen Primärschlüssel konvertiert wird.

CloudTrail -Protokolleinträge für diese Operation, die am oder nach Dezember 2022 aufgezeichnet wurden, enthalten den Schlüssel-ARN des betroffenen KMS-Schlüssels im `-responseElements.keyId`Wert, obwohl dieser Vorgang den Schlüssel-ARN nicht zurückgibt.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für `UpdatePrimaryRegion` in der Region, in der der multiregionale Schlüssel von einem Primärschlüssel in einen Replikatschlüssel (us-west-2) geändert wurde. Das `primaryRegion`-Feld zeigt die Region an, die nun den Primärschlüssel hostet (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"
}
```


Das folgende Beispiel stellt den CloudTrail Protokolleintrag für `UpdatePrimaryRegion` in der Region dar, in der der multiregionale Schlüssel von einem Replikatschlüssel in einen Primärschlüssel

(ap-northeast-1) geändert wurde. Dieser Protokolleintrag identifiziert die vorherige primäre Region nicht.


```
{
    "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

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag für die -VerifyMacOperation.


```
{
```

```
    "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"
}
```


Verify

Diese Beispiele zeigen AWS CloudTrail-Protokolleinträge für die Verify-Operation.

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag für eine Verify-Operation, die einen asymmetrischen RSA-KMS-Schlüssel verwendet, um eine digitale Signatur zu überprüfen.


```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:50:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Verify",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "messageType": "RAW"
    },
    "responseElements": null,
    "requestID": "c73ab82a-af82-4750-ae2c-b6bb790e9c28",
    "eventID": "3b4331cd-5b7b-4de5-bf5f-82ec22f0dac0",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Amazon EC2 – Beispiel 1

Das folgende Beispiel zeigt einen IAM-Prinzipal bei der Erstellung eines verschlüsselten Volumes mithilfe des Standard-Volume-Schlüssels in der Amazon-EC2-Managementkonsole.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, in dem der Benutzer Alice ein verschlüsseltes Volume mit einem Standard-Volume-Schlüssel in der Amazon EC2-Managementkonsole erstellt. Der Datensatz der EC2-Protokolldatei enthält ein `volumeId` Feld mit dem Wert `"vol-13439757"`. Der AWS KMS-Datensatz enthält ein `encryptionContext`-Feld mit dem Wert `"aws:ebs:id": "vol-13439757"`. Auch die `principalId` und `accountId` zwischen den beiden Datensätzen stimmen überein. Die Datensätze zeigen, dass bei der Erstellung eines verschlüsselten Volumes ein Datenschlüssel für die Verschlüsselung der Volume-Inhalte generiert wird.


```
{
  "Records": [
     {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
        "size": "10",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1415220618876,
        "volumeType": "gp2",
        "iops": 30,
        "encrypted": true
```

```
      },
      "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
      "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
      "eventType": "AwsApiCall",
      "recipientAccountId": "123456789012"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:19Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "&AWS; Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-13439757"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-123456789012-758241111-1415220618",
      "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
```

```
}
```

Amazon EC2 – Beispiel 2

In dem folgenden Beispiel erstellt ein IAM-Prinzipal, der eine Amazon-EC2-Instance ausführt, ein Datenvolume, das mit einem KMS-Schlüssel verschlüsselt ist. Diese Aktion generiert mehrere CloudTrail Protokolldatensätze.

Wenn das Volume erstellt wird, erhält Amazon EC2 im Auftrag des Kunden einen verschlüsselten Datenschlüssel von AWS KMS (`GenerateDataKeyWithoutPlaintext`). Dann wird eine Berechtigungserteilung (`CreateGrant`) erstellt, über die der Datenschlüssel entschlüsselt werden kann. Wenn das Volume bereitgestellt ist, ruft Amazon EC2 AWS KMS auf, um den Datenschlüssel zu entschlüsseln (`Decrypt`).

Die `instanceId` der Amazon-EC2-Instance `"i-81e2f56c"` befindet sich im `RunInstances`-Ereignis. Dieselbe Instance-ID qualifiziert den `granteePrincipal` der erstellten Berechtigungserteilung (`"111122223333:aws:ec2-infrastructure:i-81e2f56c"`) und die angenommene Rolle, die der Prinzipal im `Decrypt`-Aufruf ist (`"arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c"`).

Der Schlüssel-ARN des KMS-Schlüssels, der das Datenvolumen schützt, `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, erscheint in allen drei AWS KMS-Aufrufen (`CreateGrant`, `GenerateDataKeyWithoutPlaintext`, und `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"
    }
  ]
}
```


Überwachung mit Amazon CloudWatch

Sie können Ihre AWS KMS keys Nutzung von Amazon überwachen CloudWatch, einem AWS Service, der Rohdaten sammelt und AWS KMS in lesbare, nahezu in Echtzeit verfügbare Metriken verarbeitet. Diese Daten werden für einen Zeitraum von zwei Wochen aufgezeichnet, damit Sie auf historische Daten zugreifen und die Nutzung Ihrer KMS-Schlüssel sowie deren Änderungen besser nachvollziehen können.

Sie können Amazon verwenden CloudWatch , um Sie auf wichtige Ereignisse wie die folgenden aufmerksam zu machen.

- Das importierte Schlüsselmaterial in einem KMS-Schlüssel befindet kurz vor dem Ablaufdatum.
- Ein KMS-Schlüssel, dessen Löschung ansteht, wird immer noch verwendet.
- Das Schlüsselmaterial in einem KMS-Schlüssel wurde automatisch rotiert.
- Ein KMS-Schlüssel wurde gelöscht.

Sie können auch einen CloudWatchAmazon-Alarm einrichten, der Sie benachrichtigt, wenn Ihre Anforderungsrate einen bestimmten Prozentsatz eines Kontingents erreicht. Weitere Informationen finden Sie im AWS Sicherheitsblog unter Verwalten Sie Ihre AWS KMS API-Anforderungsraten mithilfe von Service Quotas und Amazon CloudWatch.

Themen

- AWS KMS Metriken und Dimensionen
- Metriken anzeigen AWS KMS
- CloudWatch Alarme zur Überwachung von KMS-Schlüsseln erstellen

AWS KMS Metriken und Dimensionen

AWS KMS definiert CloudWatch Amazon-Metriken, um Ihnen die Überwachung kritischer Daten und die Erstellung von Alarmen zu erleichtern. Sie können die AWS KMS Metriken mithilfe der AWS Management Console und der CloudWatch Amazon-API anzeigen.

In diesem Abschnitt werden die einzelnen AWS KMS Metriken und die Dimensionen für jede Metrik aufgeführt und einige grundlegende Anleitungen zur Erstellung von CloudWatch Alarmen auf der Grundlage dieser Metriken und Dimensionen bereitgestellt.

> **Note**
>
> Name der Dimensionsgruppe:
> Um eine Metrik in der CloudWatch Amazon-Konsole anzuzeigen, wählen Sie im Abschnitt Metriken den Namen der Dimensionsgruppe aus. Dann können Sie nach dem Metriknamen filtern. Dieses Thema enthält den Metriknamen und den Dimensionsgruppennamen für jede AWS KMS -Metrik.

Themen

- SecondsUntilKeyMaterialExpiration
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors

- XksExternalKeyManagerStates
- XksProxyLatency

SecondsUntilKeyMaterialExpiration

Die Anzahl der verbleibenden Sekunden, bis das importierte Schlüsselmaterial eines KMS-Schlüssels abläuft. Diese Metrik gilt nur für KMS-Schlüssel mit importiertem Schlüsselmaterial (Herkunft des Schlüsselmaterials `EXTERNAL`) und einem Ablaufdatum.

Verwenden Sie diese Metrik, um die Zeit zu verfolgen, die bis zum Ablauf Ihres importierten Schlüsselmaterials verbleibt. Wenn diese Zeit unter einen von Ihnen definierten Schwellenwert fällt, sollten Sie das Schlüsselmaterial mit einem neuen Ablaufdatum erneut importieren. Die `SecondsUntilKeyMaterialExpiration`-Metrik ist spezifisch für einen KMS-Schlüssel. Sie können diese Metrik nicht verwenden, um mehrere KMS-Schlüssel oder KMS-Schlüssel, die Sie möglicherweise in Zukunft erstellen werden, zu überwachen. Hilfe zur Erstellung eines CloudWatch Alarms zur Überwachung dieser Metrik finden Sie unterEinen CloudWatch Alarm für den Ablauf von importiertem Schlüsselmaterial erstellen.

Die nützlichste Statistik für diese Metrik ist `Minimum`, die Ihnen die kleinste verbleibende Zeit für alle Datenpunkte im angegebenen Statistikzeitraum angibt. Die einzige gültige Einheit für diese Metrik ist `Seconds`.

Name der Dimensionsgruppe: Per-Key Metrics (Metriken pro Schlüssel)

Dimensionen für **`SecondsUntilKeyMaterialExpiration`**

Dimension	Beschreibung; bezieht sich auf AWS
KeyId	Wert für jeden KMS-Schlüssel.

ExternalKeyStoreThrottle

Die Anzahl der Anfragen für kryptografische Operationen mit KMS-Schlüsseln in jedem externen Schlüsselspeicher, der AWS KMS gedrosselt wird (mit einem antwortet). `ThrottlingException` Diese Metrik gilt nur für externe Schlüsselspeicher.

Die `ExternalKeyStoreThrottle` Metrik gilt nur für KMS-Schlüssel in einem externen Schlüsselspeicher und nur für Anfragen für kryptografische Operationen und den Vorgang.

DescribeKey AWS KMS drosselt diese Anfragen, wenn die Anforderungsrate das benutzerdefinierte Schlüsselspeicher-Anforderungskontingent für Ihren externen Schlüsselspeicher überschreitet. Diese Metrik berücksichtigt nicht die Drosselung durch Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager.

Verwenden Sie diese Metrik, um den Wert des Anforderungskontingents für benutzerdefinierte Schlüsselspeicher zu überprüfen und anzupassen. Wenn diese Metrik darauf hindeutet, AWS KMS dass Ihre Anfragen für diese KMS-Schlüssel häufig gedrosselt werden, sollten Sie erwägen, eine Erhöhung des Kontingents für benutzerdefinierte Schlüsselspeicher-Anfragen zu beantragen. Hilfe finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Wenn Sie sehr häufig `KMSInvalidStateException`-Fehler mit einer Meldung erhalten, in der erklärt wird, dass die Anforderung „aufgrund einer sehr hohen Anforderungsrate“ abgelehnt wurde oder die Anforderung abgelehnt wurde, „weil der externe Schlüsselspeicher-Proxy nicht rechtzeitig geantwortet hat“, könnte dies darauf hinweisen, dass Ihr externer Schlüsselmanager oder externer Schlüsselspeicher-Proxy mit der aktuellen Anforderungsrate nicht Schritt halten kann. Verringen Sie wenn möglich Ihre Anforderungsrate. Sie können auch eine Verringerung des Anforderungskontingentwerts für benutzerdefinierte Schlüsselspeicher anfordern. Eine Verringerung dieses Kontingentwerts kann die Drosselung (und den `ExternalKeyStoreThrottle` Metrikwert) erhöhen, bedeutet aber, dass AWS KMS überschüssige Anfragen schnell zurückgewiesen werden, bevor sie an Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager gesendet werden. Um eine Reduzierung des Kontingents zu beantragen, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

Name der Dimensionsgruppe: Keystore Throttle Metrics (Keystore-Drossel-Metriken)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.
KmsOperation	Wert für jeden API-Vorgang AWS KMS . Diese Metrik gilt nur für kryptografische Vorgänge und den `DescribeKey` -Vorgang für KMS-Schlüssel in einem externen Schlüsselspeicher.
KeySpec	Wert für jeden Typ von KMS-Schlüssel. Die einzige unterstützte Schlüsselspezifikation für KMS-Schlüssel in einem externen Schlüsselspeicher ist SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

Die Anzahl der Tage, bis das TLS-Zertifikat für den Proxy-Endpunkt Ihres externen Schlüsselspeichers (`XksProxyUriEndpoint`) abläuft. Diese Metrik gilt nur für externe Schlüsselspeicher.

Verwenden Sie diese Metrik, um einen CloudWatch Alarm zu erstellen, der Sie über den bevorstehenden Ablauf Ihres TLS-Zertifikats informiert. Wenn das Zertifikat abläuft, AWS KMS kann nicht mit dem externen Schlüsselspeicher-Proxy kommuniziert werden. Der Zugriff auf alle durch KMS-Schlüssel geschützten Daten in Ihrem externen Schlüsselspeicher ist dann erst wieder möglich, wenn Sie das Zertifikat erneuern.

Ein Zertifikatsalarm verhindert, dass ein abgelaufenes Zertifikat den Zugriff auf Ihre verschlüsselten Ressourcen verhindert. Stellen Sie den Alarm so ein, dass Ihre Organisation Zeit hat, das Zertifikat zu erneuern, bevor es abläuft.

Name der Dimensionsgruppe: XKS Proxy Certificate Metrics (XKS-Proxy-Zertifikatsmetriken)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.
CertificateName	Subjektname (CN) im TLS-Zertifikat.

XksProxyCredentialAge

Die Anzahl der Tage, seit die aktuelle Anmeldeinformation für die Proxy-Authentifizierung des externen Schlüsselspeichers (`XksProxyAuthenticationCredential`) mit dem externen Schlüsselspeicher verknüpft wurde. Diese Zählung beginnt, wenn Sie die Anmeldeinformation für die Authentifizierung beim Erstellen oder Aktualisieren Ihres externen Schlüsselspeichers eingeben. Diese Metrik gilt nur für externe Schlüsselspeicher.

Dieser Wert soll Sie daran erinnern, wie alt Ihre Anmeldeinformation für die Authentifizierung ist. Da wir jedoch mit der Zählung beginnen, wenn Sie die Anmeldeinformation mit Ihrem externen Schlüsselspeicher verknüpfen, und nicht, wenn Sie Ihre Anmeldeinformation für die Authentifizierung auf dem Proxy des externen Schlüsselspeichers erstellen, ist dies möglicherweise kein genauer Indikator für das Alter der Anmeldeinformation auf dem Proxy.

Verwenden Sie diese Metrik, um einen CloudWatch Alarm auszulösen, der Sie daran erinnert, Ihre Proxyauthentifizierungsdaten für den externen Schlüsselspeicher zu wechseln.

Name der Dimensionsgruppe: Per-Keystore Metrics (Metriken pro Keystore)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.

XksProxyErrors

Die Anzahl der Ausnahmen im Zusammenhang mit AWS KMS Anfragen an Ihren externen Schlüsselspeicher-Proxy. Diese Anzahl umfasst Ausnahmen, zu denen der externe Schlüsselspeicher-Proxy zurückkehrt, AWS KMS und Timeoutfehler, die auftreten, wenn der externe Schlüsselspeicher-Proxy nicht AWS KMS innerhalb des Timeout-Intervalls von 250 Millisekunden reagiert. Diese Metrik gilt nur für externe Schlüsselspeicher.

Mit dieser Metrik können Sie die Fehlerquote von KMS-Schlüsseln in Ihrem externen Schlüsselspeicher verfolgen. Sie gibt Aufschluss über die häufigsten Fehler, sodass Sie Ihre Entwicklungsarbeit nach Prioritäten ordnen können. KMS-Schlüssel, die eine hohe Rate an nicht wiederholbaren Fehlern generieren, könnten beispielsweise auf ein Problem mit der Konfiguration Ihres externen Schlüsselspeichers hinweisen. Informationen zur Konfiguration Ihres externen Schlüsselspeichers finden Sie unter Anzeigen eines externen Schlüsselspeichers. Informationen zum Bearbeiten der Einstellungen Ihres externen Schlüsselspeichers finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.

Name der Dimensionsgruppe: XKS Proxy Error Metrics (XKS-Proxy-Fehlermetriken)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.
KmsOperation	Wert für jeden AWS KMS API-Vorgang, der eine Anfrage an den XKS-Proxy generiert hat.

Dimension	Beschreibung
XksOperation	Wert für jeden Proxy-API-Vorgang des externen Schlüsselspeichers.
KeySpec	Wert für jeden Typ von KMS-Schlüssel. Die einzige unterstützte Schlüsselspezifikation für KMS-Schlüssel in einem externen Schlüsselspeicher ist SYMMETRIC_DEFAULT.
ErrorType	Werte: • Wiederholbare Fehler: Wahrscheinlich vorübergehend, z. B. Netzwerkfehler. • Nicht wiederholbare Fehler: Weisen wahrscheinlich auf ein Problem mit der Konfiguration des benutzerdefinierten Schlüsselspeichers oder mit externen Komponenten hin. • N/V: Erfolgreiche Anforderung; keine Fehler
Exception Name	Werte: • Name der Ausnahme • Keine: Erfolgreiche Anforderung; keine Fehler

XksExternalKeyManagerStates

Die Anzahl der Instances des externen Schlüsselmanagers in jedem der folgenden Zustände: `Active`, `Degraded` und `Unavailable`. Die Informationen für diese Metrik stammen von dem externen Schlüsselspeicher-Proxy, der jedem externen Schlüsselspeicher zugeordnet ist. Diese Metrik gilt nur für externe Schlüsselspeicher.

Nachfolgend sind die Zustände für die Instances des externen Schlüsselmanagers aufgeführt, die mit einem externen Schlüsselspeicher verbunden sind. Jeder Proxy für einen externen Schlüsselspeicher verwendet möglicherweise andere Indikatoren zur Messung des Zustands Ihres externen Schlüsselmanagers. Weitere Informationen finden Sie in der Dokumentation Ihres externen Schlüsselspeicher-Proxys.

- `Active`: Der externe Schlüsselmanager ist fehlerfrei.
- `Degraded`: Der externe Schlüsselmanager ist fehlerhaft, kann aber trotzdem Datenverkehr bereitstellen.

- `Unavailable`: Der externe Schlüsselmanager kann keinen Datenverkehr bereitstellen.

Verwenden Sie diese Metrik, um einen CloudWatch Alarm zu erstellen, der Sie vor heruntergestuften und nicht verfügbaren externen Key-Manager-Instanzen warnt. Sehen Sie in den Protokollen Ihres externen Schlüsselspeicher-Proxys nach, welche Instances des externen Schlüsselmanagers sich in welchem Zustand befinden.

Name der Dimensionsgruppe: XKS External Key Manager Metrics (XKS-Metriken für externe Key Manager)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.
XksExternalKeyManagerState	Wert für jeden Zustand.

XksProxyLatency

Die Anzahl der Millisekunden, die ein externer Schlüsselspeicher-Proxy für die Antwort auf eine AWS KMS -Anforderung benötigt. Wenn die Anforderung eine Zeitüberschreitung aufweist, entspricht der aufgezeichnete Wert der Zeitüberschreitungsgrenze von 250 Millisekunden. Diese Metrik gilt nur für externe Schlüsselspeicher.

Verwenden Sie diese Metrik, um die Leistung Ihres externen Schlüsselspeicher-Proxys und Ihres externen Schlüsselmanagers zu bewerten. Wenn der Proxy beispielsweise bei Verschlüsselungs- und Entschlüsselungsvorgängen häufig eine Zeitüberschreitung aufweist, wenden Sie sich an Ihren externen Proxy-Administrator.

Langsame Antworten können auch darauf hindeuten, dass Ihr externer Key Manager den aktuellen Anforderungsverkehr nicht bewältigen kann. AWS KMS empfiehlt, dass Ihr externer Schlüsselmanager bis zu 1800 Anfragen für kryptografische Operationen pro Sekunde verarbeiten kann. Wenn Ihr externer Schlüsselmanager die Rate von 1 800 Anforderungen pro Sekunde nicht bewältigen kann, sollten Sie eine Verringerung Ihrer Anforderungsquote für KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher anfordern. Anforderungen für kryptografische Vorgänge,

die die KMS-Schlüssel in Ihrem externen Schlüsselspeicher verwenden, schlagen schnell mit einer Drosselungsausnahme fehl, anstatt verarbeitet und später von Ihrem externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager abgelehnt zu werden.

Name der Dimensionsgruppe: XKS Proxy Latency Metrics (XKS-Proxy-Latenzmetriken)

Dimension	Beschreibung
CustomKeyStoreId	Wert für jeden externen Schlüsselspeicher.
KmsOperation	Wert für jeden AWS KMS API-Vorgang, der eine Anfrage an den XKS-Proxy generiert hat.
XksOperation	Wert für jeden Proxy-API-Vorgang des externen Schlüsselspeichers.
KeySpec	Wert für jeden Typ von KMS-Schlüssel. Die einzige unterstützte Schlüsselspezifikation für KMS-Schlüssel in einem externen Schlüsselspeicher ist SYMMETRIC_DEFAULT.

Metriken anzeigen AWS KMS

Sie können die AWS KMS Metriken mithilfe der AWS Management Console und der CloudWatch Amazon-API anzeigen.

Um Metriken mit der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie auf der Navigationsleiste die Region aus, in der sich Ihre AWS -Ressourcen befinden.
3. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
4. Suchen Sie auf der Registerkarte Browse (Durchsuchen) nach `KMS` und wählen Sie dann KMS aus.
5. Wählen Sie den Dimensionsgruppennamen der Metrik aus, die Sie anzeigen möchten.

   Wählen Sie beispielsweise für die `SecondsUntilKeyMaterialExpiration`-Metrik die Option Per-Key Metrics (Metriken pro Schlüssel) aus.

6. Für ein Diagramm des Metrikwerts müssen Sie den Metriknamen und anschließend `Add to graph` auswählen. Um das Liniendiagramm in einen Wert zu konvertieren, wählen Sie Line (Linie)und dann Number (Zahl) aus.

So zeigen Sie Metriken mithilfe der CloudWatch Amazon-API an

Um AWS KMS Metriken mithilfe der CloudWatch API anzuzeigen, senden Sie eine ListMetricsAnfrage mit der `Namespace` Einstellung auf`AWS/KMS`. Im folgenden Beispiel wird gezeigt, wie Sie dies mit dem AWS Command Line Interface (AWS CLI) durchführen.

```
$  aws cloudwatch list-metrics --namespace AWS/KMS

{
    "Metrics": [
        {
            "Namespace": "AWS/KMS",
            "MetricName": "SecondsUntilKeyMaterialExpiration",
            "Dimensions": [
                {
                    "Name": "KeyId",
                    "Value": "1234abcd-12ab-34cd-56ef-1234567890ab"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "ExternalKeyStoreThrottle",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Encrypt"
                },
                {
                    "Name": "KeySpec",
                    "Value": "SYMMETRIC_DEFAULT"
                }
            ]
        },
```

```
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCertificateDaysToExpire",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "CertificateName",
                    "Value": "myproxy.xks.example.com"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCredentialAge",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyErrors",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                },
```

```
                {
                    "Name": "ErrorType",
                    "Value": "Retryable errors"
                },
                {
                     "Name": "ExceptionName",
                     "Value": "KMSInvalidStateException"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyHsmStates",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "XksProxyHsmState",
                    "Value": "Active"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyLatency",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                }
```

```
            ]
        }
    ]
}
```

CloudWatch Alarme zur Überwachung von KMS-Schlüsseln erstellen

Sie können einen CloudWatch Amazon-Alarm auf der Grundlage einer AWS KMS Metrik erstellen. Der Alarm sendet eine E-Mail-Nachricht, wenn ein Metrikwert einen in der Alarmkonfiguration angegebenen Schwellenwert überschreitet. Der Alarm kann die E-Mail-Nachricht an ein Amazon-SNS (Amazon Simple Notification Service)-Thema oder eine Amazon-EC2-Auto-Scaling-Richtlinie senden. Ausführliche Informationen zu CloudWatch Alarmen finden Sie unter Verwenden von CloudWatch Amazon-Alarmen im CloudWatch Amazon-Benutzerhandbuch

Erstellen eines Alarms für ablaufendes importiertes Schlüsselmaterial

Sie können die `SecondsUntilKeyMaterialExpiration`Metrik verwenden, um einen CloudWatch Alarm zu erstellen, der Sie benachrichtigt, wenn das importierte Schlüsselmaterial in einem KMS-Schlüssel bald abläuft.

Wenn Sie Schlüsselmaterial in einen KMS-Schlüssel importieren, können Sie optional festlegen, an welchem Datum und zu welcher Uhrzeit das Schlüsselmaterial abläuft. Wenn das Schlüsselmaterial abläuft, wird das Schlüsselmaterial AWS KMS gelöscht und der KMS-Schlüssel wird unbrauchbar. Wenn Sie den KMS-Schlüssel erneut nutzen möchten, müssen Sie das Schlüsselmaterial erneut importieren.

Anweisungen finden Sie unter Einen CloudWatch Alarm für den Ablauf von importiertem Schlüsselmaterial erstellen.

Erstellen eines Alarms für die Nutzung von KMS-Schlüsseln, die gelöscht werden sollen

Wenn Sie für einen KMS-Schlüssel das Löschen planen, erzwingt AWS KMS vor dem Löschen des KMS-Schlüssels eine Wartezeit. In der Wartezeit können Sie sicherzustellen, dass Sie den KMS-Schlüssel jetzt oder in der Zukunft nicht mehr benötigen. Sie können auch einen CloudWatch Alarm konfigurieren, der Sie warnt, wenn eine Person oder Anwendung während der Wartezeit versucht, den KMS-Schlüssel in einem kryptografischen Vorgang zu verwenden. Wenn Sie eine Benachrichtigung von einem solchen Alarm erhalten, können Sie das Löschen des KMS-Schlüssel abbrechen.

Anweisungen finden Sie unter Erstellen eines Alarms, der eine ausstehende Löschung eines KMS-Schlüssels erkennt.

Erstellen eines Alarms zur Überwachung eines externen Schlüsselspeichers

Sie können CloudWatch Alarme auf der Grundlage der Metriken für externe Schlüsselspeicher und KMS-Schlüssel in externen Schlüsselspeichern erstellen.

Wir empfehlen Ihnen beispielsweise, einen CloudWatch Alarm einzurichten, der Sie benachrichtigt, wenn das TLS-Zertifikat für Ihren externen Schlüsselspeicher bald abläuft (XksProxyCertificateDaysToExpire), wenn Ihr und wenn Ihr externer Schlüsselspeicher-Proxy meldet, dass sich Ihre externen Schlüsselmanager-Instanzen in einem heruntergestuften oder nicht verfügbaren Zustand befinden (XksProxyHsmStates).

Detaillierte Anweisungen finden Sie unter Überwachung eines externen Schlüsselspeichers.

Überwachung mit Amazon EventBridge

Sie können Amazon EventBridge (früher Amazon CloudWatch Events) verwenden, um Sie auf die folgenden wichtigen Ereignisse im Lebenszyklus Ihrer KMS-Schlüssel aufmerksam zu machen.

- Das Schlüsselmaterial in einem KMS-Schlüssel wurde automatisch rotiert.
- Das importierte Schlüsselmaterial in einem KMS-Schlüssel ist abgelaufen.
- Ein KMS-Schlüssel, dessen Löschung geplant war, wurde gelöscht.

AWS KMS lässt sich in Amazon integrieren EventBridge , um Sie über wichtige Ereignisse zu informieren, die sich auf Ihre KMS-Schlüssel auswirken. Jedes Ereignis wird in JSON dargestellt (JavaScriptObjektnotation) und enthält den Ereignisnamen, das Datum und die Uhrzeit des Ereignisses sowie das betroffene . Sie können diese Events erfassen und Regeln festlegen, um sie an ein oder mehrere Ziele weiterzuleiten, darunter AWS Lambda-Funktionen, Amazon-SNS-Themen, Amazon-SQS-Warteschlangen, Streams in Amazon Kinesis Data Streams oder integrierte Ziele.

Weitere Informationen zur Verwendung von EventBridge mit anderen Arten von Ereignissen, einschließlich Ereignissen, die von AWS CloudTrail bei der Aufzeichnung einer Lese-/Schreib-API-Anfrage ausgegeben werden, finden Sie im Amazon- EventBridge Benutzerhandbuch.

In den folgenden Themen werden die EventBridge Ereignisse beschrieben, die AWS KMS generiert.

KMS-CMK-Drehung

AWS KMS unterstützt Automatisches Rotieren des Schlüsselmaterials in symmetrischen KMS-Schlüsseln zur Verschlüsselung. Die jährliche Schlüsselmaterialdrehung ist optional für

Kundenverwaltete Schlüssel. Das Schlüsselmaterial für Von AWS verwaltete Schlüssel wird automatisch jedes Jahr rotiert.

Immer wenn Schlüsselmaterial AWS KMS dreht, sendet es ein `KMS CMK Rotation` Ereignis an EventBridge. AWS KMS generiert dieses Ereignis nach bestem Wissen und Gewissen.

Es folgt ein Beispiel für dieses Ereignis.

```
{
  "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"
  }
}
```

Ablauf der über KMS importierten Schlüsselinformationen

Wenn Sie Schlüsselmaterial in einen KMS-Schlüssel importieren, können Sie optional festlegen, wann das Schlüsselmaterial abläuft. Wenn das Schlüsselmaterial abläuft, AWS KMSlöscht das Schlüsselmaterial und sendet ein entsprechendes `KMS Imported Key Material Expiration` Ereignis an EventBridge. AWS KMS generiert dieses Ereignis nach bestem Wissen und Gewissen.

Es folgt ein Beispiel für dieses Ereignis.

```
{
  "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"
  }
}
```

KMS-CMK-Löschung

Wenn Sie für einen KMS-Schlüssel das Löschen planen, erzwingt AWS KMS vor dem Löschen des KMS-Schlüssels eine Wartezeit. Nach Ablauf der Wartezeit AWS KMSlöscht den KMS-Schlüssel und sendet ein `KMS CMK Deletion` Ereignis an EventBridge. AWS KMS garantiert dieses EventBridge Ereignis. Aufgrund von Wiederholungen kann es innerhalb weniger Sekunden mehrere Ereignisse generieren, die denselben KMS-Schlüssel löschen.

Es folgt ein Beispiel für dieses Ereignis.

```
{
  "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"
  }
}
```

AWS KMS Ressourcen erstellen mit AWS CloudFormation

AWS Key Management Service ist in einen Service integriert AWS CloudFormation, der Sie bei der Modellierung und Einrichtung Ihrer AWS Ressourcen unterstützt, sodass Sie weniger Zeit mit der Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur verbringen müssen. Sie erstellen eine Vorlage, die KMS-Schlüssel und Aliasnamen beschreibt, und AWS CloudFormation stellt diese Ressourcen für Sie bereit und konfiguriert sie. Informationen zur AWS KMS CloudFormation

Unterstützung von finden Sie in der Referenz zum KMS-Ressourcentyp im AWS CloudFormation Benutzerhandbuch.

Wenn Sie Ihre Vorlage verwenden AWS CloudFormation, können Sie sie wiederverwenden, um Ihre AWS KMS Ressourcen konsistent und wiederholt einzurichten. Beschreiben Sie Ihre Ressourcen einmal und stellen Sie dann dieselben Ressourcen immer wieder in mehreren AWS-Konten Regionen bereit.

Um Ressourcen für und andere AWS Dienste bereitzustellen AWS KMS und zu konfigurieren, müssen Sie AWS CloudFormation Vorlagen verstehen. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation Stacks bereitstellen möchten. Wenn Sie mit JSON oder YAML nicht vertraut sind, können Sie AWS CloudFormation Designer verwenden, um Ihnen die ersten Schritte mit Vorlagen zu erleichtern. AWS CloudFormation Weitere Informationen finden Sie unter Was ist AWS CloudFormation -Designer? im AWS CloudFormation -Benutzerhandbuch.

Regionen

AWS KMS CloudFormation Ressourcen werden in allen Regionen unterstützt, in denen dies unterstützt AWS CloudFormation wird.

AWS KMS Ressourcen in AWS CloudFormation Vorlagen

AWS KMS unterstützt die folgenden AWS CloudFormation Ressourcen.

- Die `AWS::KMS::Key`Ressource spezifiziert einen KMS-Schlüssel in AWS Key Management Service. Sie können diese Ressource verwenden, um KMS-Schlüssel mit symmetrischer Verschlüsselung, asymmetrische KMS-Schlüssel für die Verschlüsselung oder Signatur und symmetrische HMAC-KMS-Schlüssel zu erstellen. Sie können ihn verwenden`AWS::KMS::Key`, um Primärschlüssel für mehrere Regionen aller unterstützten Typen zu erstellen. Um einen multiregionalen Schlüssel zu replizieren, verwenden Sie die `AWS::KMS::ReplicaKey`-Ressource.
- `AWS::KMS::Alias` erstellt einen Alias und ordnet ihn einem KMS-Schlüssel zu. Der KMS-Schlüssel kann in der Vorlage definiert oder von einem anderen Mechanismus erstellt werden.
- `AWS::KMS::ReplicaKey` erstellt einen multiregionalen Replikatschlüssel. Verwenden Sie zum Erstellen eines multiregionalen Primärschlüssels die `AWS::KMS::Key`-Ressource. Sie können diese Ressource nicht verwenden, um multiregionale Schlüssel mit importiertem Schlüsselmaterial zu erstellen. Weitere Informationen zu multiregionalen Schlüsseln finden Sie unter Schlüssel für mehrere Regionen eingeben AWS KMS.

> **Important**
>
> Wenn Sie den Wert einer `KeyUsage`-, `KeySpec`- oder `MultiRegion`-Eigenschaft in einem vorhandenen KMS-Schlüssel ändern, wird der vorhandene KMS-Schlüssel zum Löschen vorgemerkt und es wird ein neuer KMS-Schlüssel mit dem angegebenen Wert erstellt. Sobald der vorhandene KMS-Schlüssel zum Löschen vorgemerkt ist, kann er nicht mehr genutzt werden. Wenn Sie das geplante Löschen des vorhandenen KMS-Schlüssels nicht außerhalb von abbrechen AWS CloudFormation, können alle mit dem vorhandenen KMS-Schlüssel verschlüsselten Daten nicht wiederhergestellt werden, wenn der KMS-Schlüssel gelöscht wird.

Bei den KMS-Schlüsseln, die die Vorlage erstellt, handelt es sich um tatsächliche Ressourcen in Ihrem. AWS-Konto Autorisierte Prinzipale können die von der Vorlage erstellten KMS-Schlüssel entweder mithilfe der Vorlage, der AWS KMS Konsole oder der AWS KMS APIs verwenden und verwalten. Wenn Sie einen KMS-Schlüssel aus Ihrer Vorlage löschen, wird der KMS-Schlüssel mit einer Wartezeit für das Löschen eingeplant, die Sie im Voraus angeben.

Sie können beispielsweise eine AWS CloudFormation Vorlage verwenden, um einen Test-KMS-Schlüssel mit einer Schlüsselrichtlinie, Schlüsselspezifikation, Schlüsselverwendung, Aliasen und Tags zu erstellen, die Sie bevorzugen. Sie können ihn über Ihre Testsuite ausführen, Ihre Ergebnisse überprüfen und dann die Vorlage verwenden, um den Testschlüssel für das Löschen zu planen. Später können Sie die Vorlage erneut ausführen, um einen Testschlüssel mit denselben Eigenschaften zu erstellen.

Oder Sie können eine AWS CloudFormation Vorlage verwenden, um eine bestimmte KMS-Schlüsselkonfiguration zu definieren, die Ihren Geschäftsregeln und Sicherheitsstandards entspricht. Dann können Sie diese Vorlage jederzeit verwenden, wenn Sie einen KMS-Schlüssel erstellen müssen. Sie müssen sich keine Gedanken über falsch konfigurierte Schlüssel machen. Wenn sich Ihre bevorzugte Konfiguration ändert, können Sie Ihre KMS-Schlüssel mit Ihrer Vorlage aktualisieren. Mit der Vorlage ist es beispielsweise einfach, die automatische Schlüsseldrehung für alle von der Vorlage definierten KMS-Schlüssel programmgesteuert zu aktivieren.

Weitere Informationen zu AWS KMS Ressourcen, einschließlich Beispielen, finden Sie in der Referenz zum KMS-Ressourcentyp im AWS CloudFormation Benutzerhandbuch.

Erfahren Sie mehr über AWS CloudFormation

Weitere Informationen AWS CloudFormation dazu finden Sie in den folgenden Ressourcen:

- AWS CloudFormation
- AWS CloudFormation Benutzerhandbuch
- AWS CloudFormation API Reference
- AWS CloudFormation -Benutzerhandbuch für die Befehlszeilenschnittstelle

Löschen von AWS KMS keys

Das Löschen eines AWS KMS key ist ein endgültiger und potenziell gefährlicher Vorgang. Damit löschen Sie das Schlüsselmaterial und alle Metadaten im Zusammenhang mit dem KMS-Schlüssel. Dies kann nicht rückgängig gemacht werden. Nach dem Löschen eines KMS-Schlüssels können Sie die Daten nicht mehr entschlüsseln, die mit diesem KMS-Schlüssel verschlüsselt wurden. Das bedeutet, dass die Daten nicht wiederhergestellt werden können. (Die einzigen Ausnahmen sind Replikatschlüssel für mehrere Regionen sowie asymmetrische und HMAC-KMS-Schlüssel mit importiertem Schlüsselmaterial.) Dieses Risiko ist bei asymmetrischen KMS-Schlüsseln, die zur Verschlüsselung verwendet werden, erheblich, da Benutzer ohne Warnung oder Fehler weiterhin Geheimtexte mit dem öffentlichen Schlüssel generieren können, die nicht entschlüsselt werden können, nachdem der private Schlüssel aus AWS KMS gelöscht wurde.

Löschen Sie einen KMS-Schlüssel nur, wenn Sie sicher sind, dass Sie ihn nicht mehr benötigen. Wenn Sie nicht sicher sind, sollten Sie den KMS-Schlüssel deaktivieren, anstatt ihn zu löschen. Sie können einen deaktivierten KMS-Schlüssel wieder aktivieren und die geplante Löschung eines Schlüssels abbrechen. Ein gelöschter KMS-Schlüssel kann jedoch nicht wiederhergestellt werden.

Sie können nur das Löschen eines vom Kunden verwalteten Schlüssels planen. Sie können Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel nicht löschen.

Bevor Sie einen KMS-Schlüssel löschen, möchten Sie vielleicht wissen, wie viele Chiffretexte unter diesem KMS-Schlüssel verschlüsselt wurden. AWS KMS speichert diese Information nicht und speichert keine der Chiffretexte. Um diese Informationen abzurufen, müssen Sie selbst die bisherige Nutzung eines KMS-Schlüssels feststellen. Um Hilfe zu erhalten, gehen Sie zu Feststellen der früheren Nutzung eines KMS-Schlüssels.

AWS KMS löscht Ihre KMS-Schlüssel niemals, es sei denn, Sie planen sie explizit zum Löschen und die obligatorische Wartezeit ist abgelaufen.

Für die Löschung eines KMS-Schlüssels können jedoch verschiedene Gründe sprechen:

- Der Lebenszyklus eines nicht mehr benötigten KMS-Schlüssels soll beendet werden.

- Der Aufwand und der Preis im Zusammenhang mit der Erhaltung ungenutzter KMS-Schlüssel sollen vermieden werden.
- Die Anzahl der KMS-Schlüssel, die auf das KMS-Schlüsselressourcen-Kontingent angerechnet werden, soll verringert werden.

> **Note**
>
> Wenn Sie Ihr AWS-Konto schließen, werden Ihre KMS-Schlüssel unzugänglich und sie werden Ihnen nicht mehr in Rechnung gestellt.

AWS KMS zeichnet einen Eintrag in Ihrem AWS CloudTrail-Protokoll auf, wenn Sie für den KMS-Schlüssel das Löschen planen und wenn der KMS-Schlüssel tatsächlich gelöscht wird.

Informationen zum Löschen von multiregionalen Primär- und Replikatschlüsseln finden Sie unter Löschen von multiregionalen Schlüsseln.

Themen

Über die Wartezeit

Da es sich beim Löschen eines KMS-Schlüssels um einen endgültig und potenziell gefährlichen Vorgang handelt, erfordert AWS KMS, dass Sie eine Wartezeit von 7–30 Tagen festlegen. Die Standardwartezeit beträgt 30 Tage.

Die tatsächliche Wartezeit kann jedoch bis zu 24 Stunden länger sein als die, die Sie geplant haben. Um das tatsächliche Datum und die Uhrzeit zu erhalten, zu der der KMS-Schlüssel gelöscht wird, verwenden Sie die -DescribeKeyOperation. Oder shcauen Sie in der AWS KMS-Konsole auf der

Detailseite für den KMS-Schlüssel im Abschnitt General configuration (allgemeine Konfiguration) unter Scheduled deletion date (geplantes Löschdatum). Achten Sie darauf, die Zeitzone zu notieren.

Während der Wartezeit lauten der KMS-Schlüssel-Status und der Schlüsselstatus Pending deletion (Löschung ausstehend).

- Ein KMS-Schlüssel, der zur Löschung ansteht, kann nicht in kryptografischen Operationen verwendet werden.
- AWS KMS dreht kein Schlüsselmaterial von KMS-Schlüssel, deren Löschung aussteht.

Nach Ablauf der Wartezeit, löscht AWS KMS den KMS-Schlüssel, seine Aliasse und alle zugehörigen AWS KMS-Metadaten.

Die Planung des Löschens eines KMS-Schlüssels wirkt sich möglicherweise nicht sofort auf Datenschlüssel aus, die mit dem KMS-Schlüssel verschlüsselt wurden. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Nutzen Sie die Wartezeit, um sicher zu stellen, dass Sie den KMS-Schlüssel jetzt oder in der Zukunft nicht mehr benötigen. Sie können einen Amazon CloudWatch-Alarm so konfigurieren, dass Sie gewarnt werden, wenn eine Person oder Anwendung versucht, den KMS-Schlüssel während der Wartezeit zu verwenden. Um den KMS-Schlüssel wiederherzustellen, können Sie die Löschung des Schlüssels abbrechen, bevor die Wartezeit endet. Nach dem Ende der Wartezeit können Sie die Löschung des Schlüssels nicht abbrechen. Der KMS-Schlüssel wird dann durch AWS KMS gelöscht.

Löschen asymmetrischer KMS-Schlüssel

Benutzer mit entsprechender Autorisierung können symmetrische oder asymmetrische KMS-Schlüssel löschen. Das Verfahren zum Planen des Löschens dieser KMS-Schlüssel ist für beide Arten von Schlüsseln gleich. Da jedoch der öffentliche Schlüssel eines asymmetrischen KMS-Schlüssels heruntergeladen und außerhalb von AWS KMS verwendet werden kann, stellt die Operation erhebliche zusätzliche Risiken dar, insbesondere bei asymmetrischen KMS-Schlüsseln, die für die Verschlüsselung verwendet werden (die Schlüsselnutzung ist `ENCRYPT_DECRYPT`).

- Wenn Sie das Löschen eines KMS-Schlüssels planen, ändert sich der Schlüsselstatus des KMS-Schlüssels zu Pending deletion (Löschung ausstehend) und der KMS-Schlüssel kann nicht in kryptografischen Operationen verwendet werden. Die Planung des Löschens hat jedoch keine Auswirkungen auf öffentliche Schlüssel außerhalb von AWS KMS. Benutzer, die über den öffentlichen Schlüssel verfügen, können ihn weiterhin zum Verschlüsseln von Nachrichten verwenden. Sie erhalten keine Benachrichtigung, dass sich der Schlüsselstatus geändert hat.

Wenn der Löschvorgang nicht abgebrochen wird, kann der mit dem öffentlichen Schlüssel erstellte Chiffretext nicht entschlüsselt werden.

- Alarme, Protokolle und andere Strategien, die die versuchte Verwendung eines KMS-Schlüssels erkennen, dessen Löschung aussteht, können die Verwendung des öffentlichen Schlüssels außerhalb von AWS KMS nicht erkennen.
- Wenn der KMS-Schlüssel gelöscht wird, schlagen alle AWS KMS-Aktionen mit diesem KMS-Schlüssel fehl. Benutzer, die über den öffentlichen Schlüssel verfügen, können sie jedoch weiterhin zum Verschlüsseln von Nachrichten verwenden. Diese Chiffretexte können nicht entschlüsselt werden.

Wenn Sie einen asymmetrischen KMS-Schlüssel mit der Schlüsselnutzung löschen müssen`ENCRYPT_DECRYPT`, verwenden Sie Ihre CloudTrail Protokolleinträge, um festzustellen, ob der öffentliche Schlüssel heruntergeladen und freigegeben wurde. Wenn dies der Fall ist, stellen Sie sicher, dass der öffentliche Schlüssel außerhalb von AWS KMS nicht verwendet wird. Ziehen Sie dann in Betracht, den KMS-Schlüssel zu deaktivieren, anstatt ihn zu löschen.

Das Risiko, das durch das Löschen eines asymmetrischen KMS-Schlüssels entsteht, wird für asymmetrische KMS-Schlüssel mit importiertem Schlüsselmaterial verringert. Details hierzu finden Sie unter Löschen eines KMS-Schlüssels mit importiertem Schlüsselmaterial.

Löschen von multiregionalen Schlüsseln

Benutzer, die autorisiert sind, können das Löschen von multiregionalen Primär- und Replikatschlüsseln planen. AWS KMS löscht jedoch keinen multiregionalen Primärschlüssel, der Replikatschlüssel enthält. Solange der Primärschlüssel vorhanden ist, können Sie auch einen gelöschten multiregionalen Replikatschlüssel neu erstellen. Details hierzu finden Sie unter Löschen von multiregionalen Schlüsseln.

Löschen von KMS-Schlüsseln mit importiertem Schlüsselmaterial

Autorisierte Benutzer können das Löschen von KMS-Schlüsseln mit importiertem Schlüsselmaterial planen. Durch diese Aktion werden der KMS-Schlüssel, sein Schlüsselmaterial und alle Metadaten, die dem KMS-Schlüssel zugeordnet sind, dauerhaft gelöscht.

Sie können keinen neuen KMS-Schlüssel für die symmetrische Verschlüsselung erstellen, mit dem die Geheimtexte eines gelöschten symmetrischen Verschlüsselungsschlüssels mit importiertem Schlüsselmaterial entschlüsselt werden können, selbst wenn Sie über eine Kopie des zugehörigen Schlüsselmaterials verfügen. Wenn Sie jedoch über das Schlüsselmaterial verfügen,

können Sie einen asymmetrischen KMS-Schlüssel oder HMAC-KMS-Schlüssel mit importiertem Schlüsselmaterial effektiv neu erstellen. Details hierzu finden Sie unter Löschen eines KMS-Schlüssels mit importiertem Schlüsselmaterial.

Kontrolle des Zugangs zum Löschen von Schlüsseln

Wenn Sie IAM-Richtlinien verwenden, um AWS KMS-Berechtigungen zuzulassen, sind IAM-Identitäten mit AWS-Administratorzugriff (`"Action": "*"`) oder AWS KMS-Vollzugriff (`"Action": "kms:*"`) bereits berechtigt, die Löschung von KMS-Schlüsseln zu planen und abzubrechen. Verwenden Sie die AWS KMS-Konsole oder dieAWS KMS-API, um Schlüsseladministratoren zu ermöglichen, das Löschen von Schlüsseln in der Schlüsselrichtlinie zu planen und abzubrechen.

In der Regel sind nur Schlüsseladministratoren berechtigt, das Löschen von Schlüsseln zu planen oder abzubrechen. Sie können diese Berechtigungen jedoch auch anderen IAM-Identitäten erteilen, indem Sie der Schlüsselrichtlinie oder einer IAM-Richtlinie die Berechtigungen `kms:ScheduleKeyDeletion` und `kms:CancelKeyDeletion` hinzufügen. Sie können den `kms:ScheduleKeyDeletionPendingWindowInDays` Bedingungsschlüssel auch verwenden, um die Werte, die Prinzipale im `PendingWindowInDays` Parameter einer ScheduleKeyDeletion Anforderung angeben können, weiter einzuschränken.

Schlüsseladministratoren das Planen und Abbrechen der Löschung von Schlüsseln ermöglichen (Konsole)

So erteilen Sie Schlüsseladministratoren die Berechtigung, das Löschen von Schlüsseln zu planen und abzubrechen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie den Alias oder die Schlüssel-ID des KMS-Schlüssels, dessen Berechtigungen Sie ändern möchten.
5. Wählen Sie die Registerkarte Key policy (Schlüsselrichtlinie).
6. Der nächste Schritt unterscheidet sich für die Standardansicht und die Richtlinienansicht Ihrer Schlüsselrichtlinie. Die Standardansicht ist nur verfügbar, wenn Sie die standardmäßige Konsolenschlüsselrichtlinie verwenden. Andernfalls ist nur die Richtlinienansicht verfügbar.

Wenn die Standardansicht verfügbar ist, wird auf der Registerkarte Key policy (Schlüsselrichtlinie) die Schaltfläche Switch to policy view (zur Richtlinienansicht wechseln) oder Switch to default view (zur Standardansicht wechseln) angezeigt.

- In der Standardansicht:
  - Wählen Sie unter Key deletion (Schlüssel-Löschung) die Option Allow key administrators to delete this key (Schlüsseladministratoren das Löschen dieses Schlüssels erlauben) aus.
- In der Richtlinienansicht:
  a. Wählen Sie Bearbeiten aus.
  b. Fügen Sie in der Richtlinienanweisung für Schlüsseladministratoren dem `Action`-Element die Berechtigungen `kms:ScheduleKeyDeletion` und `kms:CancelKeyDeletion` hinzu.

```
{
  "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. Wählen Sie Änderungen speichern aus.

Schlüsseladministratoren das Planen und Abbrechen der Löschung von Schlüsseln ermöglichen (AWS CLI)

Verwenden Sie die AWS Command Line Interface, um Berechtigungen für die Planung und das Abbrechen der Löschung von Schlüsseln hinzuzufügen.

So fügen Sie Berechtigungen zum Planen und Abbrechen der Löschung eines Schlüssels hinzu

1. Verwenden Sie den Befehl `aws kms get-key-policy`, um die vorhandene Schlüsselrichtlinie aufzurufen, und speichern Sie das Richtliniendokument dann in einer Datei.
2. Öffnen Sie die Richtlinien in Ihrem bevorzugten Texteditor. Fügen Sie in der Richtlinienerklärung für Schlüsseladministratoren die Berechtigungen`kms:ScheduleKeyDeletion` und `kms:CancelKeyDeletion` hinzu. Das folgende Beispiel zeigt eine Richtlinienanweisung mit diesen zwei Berechtigungen:

```
{
  "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. Verwenden Sie den Befehl `aws kms put-key-policy`, um die Schlüsselrichtlinie auf den KMS-Schlüssel anzuwenden.

Planen und Abbrechen der Löschung eines Schlüssels

Im Folgenden wird beschrieben, wie Sie die Löschung von einzelregionalen AWS KMS keys (KMS-Schlüssel) in AWS KMS mittels der AWS Management Console, der AWS CLI und dem AWS SDK for Java planen und abbrechen können.

Informationen zum Planen des Löschens von multiregionalen Schlüsseln finden Sie unter Löschen von multiregionalen Schlüsseln.

> **Warning**
>
> Das Löschen eines KMS-Schlüssels in KMS ist ein endgültiger und potenziell gefährlicher Vorgang. Fahren Sie nur fort, wenn Sie sicher sind, dass Sie den KMS-Schlüssel später nicht mehr verwenden müssen. Wenn Sie nicht sicher sind, sollten Sie den KMS-Schlüssel deaktivieren, anstatt ihn zu löschen.

Bevor Sie einen KMS-Schlüssel löschen können, müssen Sie über die entsprechende Berechtigung verfügen. Hinweise zum Erteilen dieser Berechtigungen an Schlüsseladministratoren finden Sie unter Kontrolle des Zugangs zum Löschen von Schlüsseln. Sie können den Bedingungsschlüssel `kms:ScheduleKeyDeletionPendingWindowInDays` auch verwenden, um die Wartezeit weiter einzuschränken, z. B. um eine Mindestwartezeit durchzusetzen.

AWS KMS zeichnet einen Eintrag in Ihrem AWS CloudTrail-Protokoll auf, wenn Sie für den KMS-Schlüssel das Löschen planen und wenn der KMS-Schlüssel tatsächlich gelöscht wird.

Planen und Abbrechen der Löschung eines Schlüssels (Konsole)

In der AWS Management Console können Sie das Löschen mehrerer KMS-Schlüssel gleichzeitig planen und abbrechen.

So planen Sie die Löschung

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.

Sie können das Löschen von Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel nicht planen.

4. Aktivieren Sie das Kontrollkästchen neben dem KMS-Schlüssel, den Sie löschen möchten.
5. Wählen Sie Key actions (Schlüsselaktionen), Schedule key deletion (Schlüssellöschung planen).
6. Lesen und berücksichtigen Sie die Warnung und die Informationen zum Abbrechen des Löschens während der Wartezeit. Wenn Sie den Löschvorgang abbrechen möchten, wählen Sie unten auf der Seite Cancel (Abbrechen).
7. Geben Sie für Waiting period (in days) (Wartezeit (in Tagen)) eine Anzahl von Tagen zwischen 7 und 30 ein.
8. Überprüfen Sie die KMS-Schlüssel, die Sie löschen.
9. Aktivieren Sie zur Bestätigung, dass Sie den Schlüssel löschen möchten, das Kontrollkästchen neben Confirm that you want to delete this key in ***<number of days>*** days..
10. Wählen Sie Schedule deletion.

Der Status des Schlüssels wechselt zu Pending Deletion (Löschung ausstehend).

So brechen Sie die Löschung eines Schlüssels ab

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Aktivieren Sie das Kontrollkästchen neben dem KMS-Schlüssel, den Sie wiederherstellen möchten.
5. Wählen Sie Key actions (Schlüsselaktionen), Cancel key deletion (Schlüssellöschung abbrechen).

Der Status des KMS-Schlüssels wechselt von Pending Deletion (Löschung ausstehend zu Disabled (deaktiviert). Um den KMS-Schlüssel verwenden zu können, müssen Sie ihn aktivieren.

Planen und Abbrechen der Löschung eines Schlüssels (AWS CLI)

Verwenden Sie den Befehl `aws kms schedule-key-deletion`, um die Löschung eines vom Kunden verwalteten Schlüssels, wie im folgenden Beispiel gezeigt, zu planen.

Sie können das Löschen eines Von AWS verwalteter Schlüssel oder AWS-eigener Schlüssel nicht planen.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --
pending-window-in-days 10
```

Wenn der Befehl erfolgreich verwendet wird, gibt die AWS CLI eine Ausgabe ähnlich der im folgenden Beispiel dargestellten zurück:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": 1598304792.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 10
}
```

Verwenden Sie den Befehl `aws kms cancel-key-deletion`, um die Löschung eines Schlüssels, wie im folgenden Beispiel gezeigt, mithilfe der AWS CLI abzubrechen.

```
$ aws kms cancel-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Wenn der Befehl erfolgreich verwendet wird, gibt die AWS CLI eine Ausgabe ähnlich der im folgenden Beispiel dargestellten zurück:

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

Der Status des KMS-Schlüssels wechselt von Pending deletion (Löschung aussthend) zu Disabled (deaktiviert). Um den KMS-Schlüssel verwenden zu können, müssen Sie ihn aktivieren.

Planen und Abbrechen der Löschung eines Schlüssels (AWS SDK for Java)

Das folgende Beispiel zeigt, wie Sie das Löschen eines vom Kunden verwalteten Schlüssels mit dem AWS SDK for Java planen. Dieses Beispiel erfordert die vorherige Instanziierung eines `AWSKMSClient` als `kms`.

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

int PendingWindowInDays = 10;

ScheduleKeyDeletionRequest scheduleKeyDeletionRequest =
new
 ScheduleKeyDeletionRequest().withKeyId(KeyId).withPendingWindowInDays(PendingWindowInDays);
kms.scheduleKeyDeletion(scheduleKeyDeletionRequest);
```

Das folgende Beispiel veranschaulicht das Abbrechen der Löschung eines Schlüssels mit dem AWS SDK for Java. Dieses Beispiel erfordert die vorherige Instanziierung eines `AWSKMSClient` als `kms`.

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

CancelKeyDeletionRequest cancelKeyDeletionRequest =
new CancelKeyDeletionRequest().withKeyId(KeyId);
kms.cancelKeyDeletion(cancelKeyDeletionRequest);
```

Der Status des KMS-Schlüssels wechselt von Pending deletion (Löschung ausstehend) zu Disabled (deaktiviert). Um den KMS-Schlüssel verwenden zu können, müssen Sie ihn aktivieren.

Erstellen eines Alarms, der eine ausstehende Löschung eines KMS-Schlüssels erkennt

Sie können die Funktionen von AWS CloudTrail, Amazon CloudWatch Logs und Amazon Simple Notification Service (Amazon SNS) kombinieren, um einen Amazon- CloudWatch Alarm zu erstellen, der Sie benachrichtigt, wenn ein Benutzer in Ihrem Konto versucht, einen KMS-Schlüssel zu verwenden, dessen Löschung aussteht. Wenn Sie diese Benachrichtigung erhalten, sollten Sie eventuell das Löschen des KMS-Schlüssels stornieren und erneut abwägen, ob es sinnvoll ist, diesen KMS-Schlüssel zu löschen.

Mit den folgenden Verfahren wird ein Alarm erstellt, der Sie benachrichtigt, wenn die Fehlermeldung „*Key ARN* `is pending deletion`“ in Ihre CloudTrail Protokolldateien geschrieben wird. Diese Fehlermeldung gibt an, dass ein Benutzer oder eine Anwendung versucht hat, den KMS-Schlüssel in einer kryptografischen Operation zu verwenden. Da diese Benachrichtigung mit der Fehlermeldung verknüpft ist, wird sie nicht ausgelöst, wenn Sie API-Operationen ausführen, die für KMS-Schlüssel zulässig sind, bei denen die Löschung aussteht, z. B. `ListKeys`, `CancelKeyDeletion` und

`PutKeyPolicy`. Eine Liste der AWS KMS-API-Operationen, für die diese Fehlermeldung zurückgegeben wird, finden Sie unter Wichtige Zustände von AWS KMS Schlüsseln.

Die Benachrichtigungs-E-Mail, die Sie erhalten, enthält weder den KMS-Schlüssel noch die kryptografische Operation. Diese Informationen finden Sie in Ihrem CloudTrail-Protokoll. Die E-Mail informiert Sie darüber, dass der Alarmstatus von OK zu Alarm geändert wurde. Weitere Informationen zu CloudWatch Alarmen und Statusänderungen finden Sie unter Verwenden von Amazon CloudWatch-Alarmen im Amazon- CloudWatch Benutzerhandbuch.

> **Warning**
>
> Dieser Amazon- CloudWatch Alarm kann die Verwendung des öffentlichen Schlüssels eines asymmetrischen KMS-Schlüssels außerhalb von nicht erkennenAWS KMS. Weitere Informationen zu den besonderen Risiken des Löschens asymmetrischer KMS-Schlüssel, die für die Kryptographie von öffentlichen Schlüsseln verwendet werden, einschließlich der Erstellung von Chiffretexten, die nicht entschlüsselt werden können, finden Sie unter Löschen asymmetrischer KMS-Schlüssel.

Themen

- Anforderungen für einen CloudWatch Alarm
- Erstellen des CloudWatch Alarms

Anforderungen für einen CloudWatch Alarm

Bevor Sie einen CloudWatch Alarm erstellen, müssen Sie einen -AWS CloudTrailTrail erstellen und so konfigurieren, CloudTrail dass CloudTrail Protokolldateien an Amazon CloudWatch Logs übermittelt werden. Sie benötigen auch ein Amazon-SNS-Thema für die Alarmbenachrichtigung.

- Erstellen eines CloudTrail-Pfads.

  CloudTrail wird automatisch auf Ihrem aktiviertAWS-Konto, wenn Sie das Konto erstellen. Wenn Sie jedoch einen fortlaufenden Datensatz der Ereignisse in Ihrem Konto, einschließlich der Ereignisse für AWS KMS benötigen, müssen Sie einen Trail erstellen.
- Konfigurieren Sie CloudTrail , um Ihre CloudWatch Protokolldateien bereitzustellen.

Konfigurieren Sie die Bereitstellung Ihrer CloudTrail Protokolldateien an - CloudWatch Protokolle. Auf diese Weise kann CloudWatch Logs die Protokolle auf AWS KMS API-Anforderungen überwachen, die versuchen, einen KMS-Schlüssel zu verwenden, dessen Löschung aussteht.

- Erstellen Sie ein Amazon-SNS-Thema.

  Wenn Ihr Alarm ausgelöst wird, werden Sie benachrichtigt, indem eine E-Mail-Nachricht an eine E-Mail-Adresse in einem Amazon Simple Notification Service (Amazon SNS)-Thema gesendet wird.

Erstellen des CloudWatch Alarms

In diesem Verfahren erstellen Sie einen CloudWatch Protokollgruppen-Metrikfilter, der Instances der Ausnahme „Ausstehendes Löschen“ findet. Anschließend erstellen Sie einen CloudWatch Alarm basierend auf der Protokollgruppenmetrik. Informationen zu Protokollgruppen-Metrikfiltern finden Sie unter Erstellen von Metriken aus Protokollereignissen mithilfe von Filtern im Amazon- CloudWatch Logs-Benutzerhandbuch.

1. Erstellen Sie einen CloudWatch Metrikfilter, der CloudTrail Protokolle analysiert.

   Folgen Sie den Anweisungen unter Create a metric filter for a log group (Erstellen eines Metrikfilters für eine Protokollgruppe) mit den folgenden erforderlichen Werten. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Filtermuster	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
Metrikwert	1

2. Erstellen Sie einen CloudWatch Alarm basierend auf dem Metrikfilter, den Sie in Schritt 1 erstellt haben.

   Folgen Sie den Anweisungen unter Erstellen eines CloudWatch Alarms basierend auf einem Protokollgruppen-Metrikfilter unter Verwendung der folgenden erforderlichen Werte. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrikfilter	Der Name des Metrikfilters, den Sie in Schritt 1 erstellt haben.
Threshold-Typ	Statisch
Bedingungen	Immer, wenn *metric-name* größer als 1 ist
Datenpunkte für Alarm	1 von 1
Fehlende Datenbehandlung	Fehlende Daten als gut behandeln (keine Verletzung des Schwellenwerts)

Nachdem Sie dieses Verfahren abgeschlossen haben, erhalten Sie jedes Mal eine Benachrichtigung, wenn Ihr neuer CloudWatch Alarm in den -ALARMStatus wechselt. Wenn Sie eine Benachrichtigung für diesen Alarm erhalten, kann das bedeuten, dass zum Verschlüsseln oder Entschlüsseln von Daten immer noch ein KMS-Schlüssel erforderlich ist, der gelöscht werden soll. In diesem Fall stornieren Sie das Löschen des KMS-Schlüssels und überdenken Sie, ob es sinnvoll ist, diesen KMS-Schlüssel zu löschen.

Feststellen der früheren Nutzung eines KMS-Schlüssels

Bevor Sie einen KMS-Schlüssel löschen, möchten Sie vielleicht wissen, wie viele Chiffretexte unter diesem KMS-Schlüssel verschlüsselt wurden. AWS KMS speichert diese Informationen nicht und speichert keine der Chiffretexte. Wenn Sie wissen, wie ein KMS-Schlüssel in der Vergangenheit verwendet wurde, kann das Ihnen bei der Entscheidung helfen, ob Sie ihn in der Zukunft benötigen. In diesem Thema werden verschiedene Strategien vorgeschlagen, mit denen Sie die frühere Nutzung eines KMS-Schlüssels feststellen können.

> **Warning**
>
> Diese Strategien zur Bestimmung vergangener und tatsächlicher Nutzung sind nur für AWS-Benutzer und AWS KMS-Produktionen wirksam. Sie können die Nutzung des öffentlichen Schlüssels eines asymmetrischen KMS-Schlüssels außerhalb von AWS KMS nicht erkennen. Weitere Informationen zu den besonderen Risiken des Löschens asymmetrischer KMS-

Schlüssel, die für die Kryptographie von öffentlichen Schlüsseln verwendet werden, einschließlich der Erstellung von Chiffretexten, die nicht entschlüsselt werden können, finden Sie unter Löschen asymmetrischer KMS-Schlüssel.

Themen

- Untersuchen der KMS-Schlüssel-Berechtigungen, um den Umfang einer potentiellen Nutzung zu bestimmen
- Untersuchen der AWS CloudTrail-Protokolle, um die tatsächliche Nutzung zu bestimmen

Untersuchen der KMS-Schlüssel-Berechtigungen, um den Umfang einer potentiellen Nutzung zu bestimmen

Wenn Sie bestimmen, wer oder was derzeit Zugriff auf einen KMS-Schlüssel hat, kann Ihnen das helfen zu bestimmen, wie weit der KMS-Schlüssel verwendet wurde und ob er noch benötigt wird. Um zu erfahren, wie man feststellt, wer oder was derzeit Zugriff auf einem KMS-Schlüssel hat, öffnen Sie Bestimmen des Zugriffs auf AWS KMS keys.

Untersuchen der AWS CloudTrail-Protokolle, um die tatsächliche Nutzung zu bestimmen

Sie können mit der Nutzungshistorik eines KMS-Schlüssels feststellen, ob Chiffretexte unter einem bestimmten KMS-Schlüssel verschlüsselt sind.

Alle AWS KMS-API-Aktivitäten werden in AWS CloudTrail-Protokolldateien aufgezeichnet. Wenn Sie in der Region, in der sich Ihr KMS-Schlüssel befindet, einen CloudTrail Trail erstellt haben, können Sie Ihre CloudTrail Protokolldateien untersuchen, um einen Verlauf aller AWS KMS API-Aktivitäten für einen bestimmten KMS-Schlüssel anzuzeigen. Wenn Sie keinen Trail haben, können Sie die letzten Ereignisse trotzdem in Ihrem CloudTrail Ereignisverlauf anzeigen. Weitere Informationen zur AWS KMS Verwendung von CloudTrailfinden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

Die folgenden Beispiele zeigen CloudTrail Protokolleinträge, die generiert werden, wenn ein KMS-Schlüssel zum Schutz eines in Amazon Simple Storage Service (Amazon S3) gespeicherten Objekts verwendet wird. In diesem Beispiel wird das Objekt unter Verwendung der serverseitigen Verschlüsselung mit KMS-Schlüsseln (SSE-KMS) in Simple Storage Service (Amazon S3) hochgeladen. Wenn Sie ein Objekt auf Simple Storage Service (Amazon S3) mit SSE-KMS

hochladen, geben Sie den KMS-Schlüssel an, mit dem das Objekts geschützt werden soll. Amazon S3 verwendet die -AWS KMSGenerateDataKeyOperation, um einen eindeutigen Datenschlüssel für das Objekt anzufordern, und dieses Anforderungsereignis wird CloudTrail mit einem Eintrag ähnlich dem folgenden angemeldet:


```
{
  "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"
}
```

Wenn Sie dieses Objekt später von Simple Storage Service (Amazon S3) herunterladen, sendet Amazon S3 eine `Decrypt`-Anforderung an AWS KMS, um den Datenschlüssel des Objekts mit dem angegebenen KMS-Schlüssel zu entschlüsseln. Wenn Sie dies tun, enthalten Ihre CloudTrail Protokolldateien einen Eintrag ähnlich dem folgenden:

```
{
  "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"
}
```

Die gesamte AWS KMS-API-Aktivität wird von CloudTrail protokolliert. Durch die Auswertung dieser Protokolleinträge können Sie die bisherige Nutzung eines bestimmten KMS-Schlüssels feststellen, und so können Sie bestimmen, ob Sie ihn löschen möchten.

Weitere Beispiele dafür, wie APIAWS KMS-Aktivitäten in Ihren CloudTrail Protokolldateien angezeigt werden, finden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail. Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail -Benutzerhandbuch.

Wichtige Zustände von AWS KMS Schlüsseln

An hat AWS KMS key immer einen Schlüsselstatus. Operationen auf dem KMS-Schlüssel und seiner Umgebung können diesen Schlüsselstatus entweder vorübergehend ändern oder bis eine andere Operation seinen Schlüsselstatus ändert.

Die Tabelle in diesem Abschnitt zeigt, wie sich wichtige Status auf Aufrufe von AWS KMS API-Vorgängen auswirken. Aufgrund des Schlüsselstatus wird erwartet, dass eine Operation für einen KMS-Schlüssel erfolgreich ist (#), fehlschlägt (X) oder nur unter bestimmten Bedingungen erfolgreich ist (?). Bei KMS-Schlüssel mit importiertem Schlüsselmaterial kommt es häufig zu abweichenden Ergebnissen.

Diese Tabelle enthält nur die API-Operationen, die einen vorhandenen KMS-Schlüssel verwenden. Andere Operationen, wie CreateKeyund ListKeys, werden weggelassen.

Themen

- Schlüsselstatus und KMS-Schlüsseltypen

- Schlüsselstatus-Tabelle

Schlüsselstatus und KMS-Schlüsseltypen

Der Typ des KMS-Schlüssels bestimmt den Schlüsselstatus, den er haben kann.

- Alle KMS-Schlüssel können im Status `Enabled`, `Disabled`, und `PendingDeletion` sein.
- Die meisten KMS-Schlüssel werden im Status `Enabled` erstellt. Schlüssel mit importiertem Schlüsselmaterial werden im Status `PendingImport` erstellt.
- Der Status `PendingImport` gilt nur für KMS-Schlüssel mit importiertem Schlüsselmaterial.
- Der Status `Unavailable` gilt nur für KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher. Ein KMS-Schlüssel in einem AWS CloudHSM Schlüsselspeicher liegt vor`Unavailable`, wenn der benutzerdefinierte Schlüsselspeicher absichtlich von seinem AWS CloudHSM Cluster getrennt wird. Ein KMS-Schlüssel in einem externen Schlüsselspeicher ist `Unavailable`, wenn der benutzerdefinierte Schlüsselspeicher von seinem externen Schlüsselspeicher-Proxy absichtlich getrennt wird. Sie können diese nicht-verfügbaren KMS-Schlüssel anzeigen und verwalten, Sie können sie jedoch nicht in kryptografischen Operationen verwenden.

  Der Schlüsselstatus eines KMS-Schlüssels in einem benutzerdefinierten Schlüsselspeicher wird durch Änderungen an seinem Unterstützungsschlüssel nicht beeinflusst. Ein KMS-Schlüssel in einem AWS CloudHSM Schlüsselspeicher wird durch Änderungen an seinem zugehörigen Schlüsselmaterial im AWS CloudHSM Cluster nicht beeinträchtigt. Ein KMS-Schlüssel in einem externen Schlüsselspeicher wird durch Änderungen an seinem externen Schlüssel in einem externen Schlüsselmanager nicht beeinflusst. Wenn der Unterstützungsschlüssel deaktiviert oder gelöscht wird, ändert sich der Status des KMS-Schlüssels nicht. Kryptografische Vorgänge, die den KMS-Schlüssel verwenden, schlagen jedoch fehl.
- Der `Creating`-, `Updating`-, und `PendingReplicaDeletion`-Schlüsselstatus gilt nur für multiregionale Schlüssel.
  - Ein multiregionaler Replikatschlüssel befindet sich während der Erstellung im vorübergehenden Schlüsselstatus `Creating`. Dieser Vorgang ist möglicherweise noch im Gange, wenn der ReplicateKeyVorgang abgeschlossen ist. Wenn der Replikationsprozess abgeschlossen ist, befindet sich der Replikatschlüssel im Status `Enabled` oder `PendingImport`.
  - Multiregionale Schlüssel befinden sich im vorübergehenden Schlüsselstatus `Updating`, während die primäre Region aktualisiert wird. Dieser Vorgang ist möglicherweise noch im Gange,

wenn der UpdatePrimaryRegionVorgang abgeschlossen ist. Wenn der Aktualisierungsvorgang abgeschlossen ist, setzen die Primär- und Replikatschlüssel den `Enabled`-Schlüsselstatus fort.

- Wenn Sie das Löschen eines multiregionalen Primärschlüssels planen, der Replikatschlüssel besitzt, befindet sich der Primärschlüssel im Status `PendingReplicaDeletion`, bis alle seine Replikatschlüssel gelöscht werden. Danach wechselt der Schlüsselstatus zu `PendingDeletion`. Details hierzu finden Sie unter Löschen von multiregionalen Schlüsseln.

Schlüsselstatus-Tabelle

Die folgende Tabelle zeigt, wie sich der Schlüsselstatus eines KMS-Schlüssels auf AWS KMS - Operationen auswirkt.

Die Beschreibungen der nummerierten Fußnoten ([n] sind am Ende dieses Themas.

> **Note**
>
> Möglicherweise müssen Sie horizontal oder vertikal Scrollen, um alle Daten in dieser Tabelle anzuzeigen.

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
CancelKeyDeletion	✗ [4]	✗ [4]	✓	✗ [4]	✗ [4], [13]	✗ [4]	✗ [4]
CreateAlias	✓	✓	✗	✓	✓	✓	✓

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
			[3]				
CreateGrant	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	✓	⊗ [14]	✓
Decrypt	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
DeleteAlias	✓	✓	✓	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓ (keine Auswirkung)	N/A	⊗ [14]	⊗ [15]
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
DisableKeyRotation	 [7]	 [1] oder [7]	 [3] oder [7]	 [6]	 [7]	 [14]	 [7]
EnableKey			 [3]	 [5]	 [12]	 [14]	 [15]
EnableKeyRotation	 [7]	 [1] oder [7]	 [3] oder [7]	 [6]	 [7]	 [14]	 [7]
Encrypt		 [1]	 [2] oder [3]	 [5]	 [11]	 [14]	
GenerateDataKey		 [1]	 [2] oder [3]	 [5]	 [11]	 [14]	

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
GenerateDataKeyPair	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateMac	✓	⊗ [1]	⊗ [2] oder [3]	N/A	N/A	⊗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	? [7]	? [7]	? [7]	⊗ [6]	⊗ [7]	? [7]	? [7]

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
GetParametersForImport	 [9]	 [9]	 [8] oder [9]		 [9]	 [14]	 [15]
GetPublicKey		 [1]	 [2] oder [3]	N/A	N/A	 [14]	
ImportKeyMaterial	 [9]	 [9]	 [8] oder [9]		 [9]	 [14]	
ListAliases							
ListGrants							
ListKeyPolicies							

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
ListKeyRotations	 [7]	 [7]	 [7]	 [6]	 [7]	 [7]	 [7]
ListResourceTags							
PutKeyPolicy							
ReEncrypt		 [1]	 [2] oder [3]	 [5]	 [11]	 [14]	
ReplicateKey		 [1]	 [2] oder [3]	 [5]	N/A	 [14]	 [15]
RetireGrant							
RevokeGrant							

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
RotateKeyOnDemand	[7]	[1] oder [7]	[3] oder [7]	[6]	[7]	[14]	[7]
ScheduleKeyDeletion	✓	✓	[3]	✓	✓	✓	[15]
Sign	✓	[1]	[2] oder [3]	N/A	N/A	[14]	✓
TagResource	✓	✓	[3]	✓	✓	✓	✓
UntagResource	✓	✓	[3]	✓	✓	✓	✓
UpdateAlias	✓	✓	[10]	✓	✓	✓	✓

API	Enabled	Disabled	Löschen ausstehend Löschen des Replikats ausstehend	Import ausstehend	Nicht verfügbar	Erstellen	Aktualisieren
UpdateKeyDescription	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊗ [1]	⊗ [2] oder [3]	⊗ [5]	N/A	⊗ [14]	✓
Verify	✓	⊗ [1]	⊗ [2] oder [3]	N/A	N/A	⊗ [14]	✓
VerifyMac	✓	⊗ [1]	⊗ [2] oder [3]	N/A	N/A	⊗ [14]	✓

Tabellendetails

- [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] Wenn der KMS-Schlüssel importiertes Schlüsselmaterial enthält oder sich in einem benutzerdefinierten Schlüsselspeicher befindet: `UnsupportedOperationException`.
- [8] Wenn der KMS-Schlüssel importiertes Schlüsselmaterial enthält: `KMSInvalidStateException`
- [9] Wenn der KMS-Schlüssel kein importiertes Schlüsselmaterial enthält bzw. enthalten kann: `UnsupportedOperationException`
- [10] Wenn für den Quell-KMS-Schlüssel die Löschung aussteht, wird der Befehl erfolgreich ausgeführt. Wenn für den Ziel-KMS-Schlüssel die Löschung aussteht, schlägt der Befehl mit diesem Fehler fehl: `KMSInvalidStateException : <key ARN> is pending deletion.`
- [11] `KMSInvalidStateException: <key ARN> is unavailable.` Sie können diese Operation nicht auf einem KMS-Schlüssel durchführen, der nicht verfügbar ist.
- [12] Die Operation ist erfolgreich, aber der Schlüsselstatus des KMS-Schlüssels ändert sich erst, wenn er tatsächlich verfügbar ist.
- [13] Wenn ein KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zur Löschung aussteht, bleibt der Schlüsselstatus auf `PendingDeletion`, auch wenn der KMS-Schlüssel nicht mehr verfügbar ist. Dies erlaubt es Ihnen, die Löschung des KMS-Schlüssels vor Ablauf der Wartefrist jederzeit abzubrechen.
- [14] `KMSInvalidStateException: <key ARN> is creating.` AWS KMS löst diese Ausnahme aus, während ein Schlüssel für mehrere Regionen repliziert wird (). `ReplicateKey`
- [15] `KMSInvalidStateException: <key ARN> is updating.` AWS KMS löst diese Ausnahme aus, während die primäre Region eines Schlüssels mit mehreren Regionen aktualisiert wird (). `UpdatePrimaryRegion`

Authentifizierung und Zugriffskontrolle für AWS KMS

Um AWS KMS zu verwenden, müssen Sie über Anmeldeinformationen verfügen, die AWS zur Authentifizierung Ihrer Anforderungen verwenden kann. Die Anmeldedaten müssen die Berechtigung zum Zugriff auf AWS-Ressourcenenthalten: AWS KMS keys und Aliase. Kein AWS-Prinzipal hat Berechtigungen für einen KMS-Schlüssel, sofern diese Berechtigung nicht explizit erteilt und niemals verweigert wird. Es gibt keine implizite oder automatische Berechtigung zur Verwendung oder Verwaltung eines KMS-Schlüssels.

Die wichtigste Möglichkeit zum Verwalten des Zugriffs auf Ihre AWS KMS-CMKs besteht in der Anwendung von Richtlinien. Richtlinien sind Dokumente, in denen beschrieben wird, welche Prinzipale auf welche Ressourcen zugreifen können. Richtlinien, die einer IAM-Identität angefügt sind, werden als identitätsbasierte Richtlinien (oder IAM-Richtlinien) bezeichnet, während Richtlinien, die anderen Arten von Ressourcen zugeordnet sind, als Ressourcenrichtlinien bezeichnet werden. AWS KMS-Ressourcenrichtlinien für KMS-Schlüssel werden als Schlüsselrichtlinien bezeichnet. Alle KMS-Schlüssel verfügen über eine Schlüsselrichtlinie.

Um den Zugriff auf Ihre AWS KMS-Aliase zu steuern, verwenden Sie IAM-Richtlinien. Damit Prinzipale Aliase erstellen können, müssen Sie die Berechtigung für den Alias in einer IAM-Richtlinie und die Berechtigung für den Schlüssel in einer Schlüsselrichtlinie erteilen. Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.

Um den Zugriff auf Ihre KMS-Schlüssel zu steuern, können Sie die folgenden Richtlinienmechanismen verwenden.

- Schlüsselrichtlinie – Jeder KMS-Schlüssel besitzt eine Schlüsselrichtlinie. Schlüsselrichtlinien sind der primäre Mechanismus zur Steuerung des Zugriffs auf KMS-Schlüssel. Sie können einzig die Schlüsselrichtlinie zum Steuern des Zugriffs verwenden, was bedeutet, dass der vollständige Umfang des Zugriffs auf den KMS-Schlüssel in einem einzigen Dokument definiert wird (der Schlüsselrichtlinie). Weitere Informationen zur Verwendung von Schlüsselrichtlinien finden Sie unter Schlüsselrichtlinien.
- Verwendung von IAM-Richtlinien in Kombination mit der Schlüsselrichtlinie – Sie können IAM-Richtlinien in Kombination mit der Schlüsselrichtlinie verwenden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Diese Art der Zugriffssteuerung ermöglicht Ihnen die Verwaltung aller Berechtigungen für Ihre IAM-Identitäten in IAM. Um mit einer IAM-Richtlinie den Zugriff auf einen KMS-Schlüssel zu ermöglichen, muss die Schlüsselrichtlinie dies explizit erlauben. Weitere Informationen zur Verwendung von IAM-Richtlinien finden Sie unter IAM-Richtlinien.

- Erteilungen – Sie können Erteilungen in Kombination mit der Schlüsselrichtlinie und IAM-Richtlinien verwenden, um den Zugriff auf einen KMS-Schlüssel zu ermöglichen. Diese Art der Zugriffssteuerung ermöglicht es Ihnen, Zugriff auf den KMS-Schlüssel in der Schlüsselrichtlinie zu gewähren und Identitäten die Berechtigung zur Delegierung ihres Zugriffs zu erteilen. Weitere Informationen zur Verwendung von Zugriffserteilungen finden Sie unter Erteilungen in AWS KMS.

KMS-Schlüssel gehören zu dem AWS-Konto, in dem sie erstellt wurden. Keine Identität und kein Prinzipal, einschließlich des AWS-Konto-Root-Benutzers, hat die Berechtigung, einen KMS-Schlüssel zu verwenden oder zu verwalten, sofern diese Berechtigung nicht ausdrücklich in einer Schlüsselrichtlinie, IAM-Richtlinie oder einer Berechtigungserteilung bereitgestellt wird. Die IAM-Identität, die einen KMS-Schlüssel erstellt, gilt nicht als Schlüsselbesitzer und hat nicht automatisch die Berechtigung, den von ihm erstellten KMS-Schlüssel zu verwenden oder zu verwalten. Wie jede andere Identität muss der Schlüsselersteller eine Berechtigung über eine Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung erhalten. Identitäten, die die `kms:CreateKey`-Berechtigung haben, können jedoch die ursprüngliche Schlüsselrichtlinie festlegen und sich selbst die Berechtigung zur Verwendung oder Verwaltung des Schlüssels erteilen.

In den folgenden Themen erfahren Sie, wie Sie Ihre Ressourcen mithilfe von AWS Identity and Access Management (IAM)- und AWS KMS-Berechtigungen schützen können, indem Sie den Zugriff darauf kontrollieren.

Themen

- Testen der Berechtigungen

Konzepte der AWS KMS-Zugriffskontrolle

Lernen Sie die Konzepte kennen, die in Diskussionen über die Zugangskontrolle AWS KMS verwendet werden.

Themen

- Authentifizierung
- Autorisierung
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- AWS KMS-Ressourcen

Authentifizierung

Authentifizierung ist der Prozess der Überprüfung Ihrer Identität. Um eine Anfrage an AWS KMS zu senden, müssen Sie sich bei AWS mit Ihren AWS-Anmeldeinformationen anmelden.

Autorisierung

Die Autorisierung berechtigt zum Senden von Anfragen zum Erstellen, Verwalten oder Verwenden von AWS KMS-Ressourcen. Sie müssen beispielsweise autorisiert sein, um einen KMS-Schlüssel in einer kryptografischen Operation zu verwenden.

Verwenden Sie Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen, um den Zugriff auf Ihre AWS KMS-Ressourcen zu steuern. Jeder KMS-Schlüssel muss über eine Schlüsselrichtlinie verfügen. Wenn die Schlüsselrichtlinie es zulässt, können Sie auch IAM-Richtlinien und Erteilungen verwenden, um Prinzipalen Zugriff auf den KMS-Schlüssel zu geben. Um Ihre Autorisierung zu verfeinern, können Sie Bedingungsschlüssel verwenden, die den Zugriff nur dann erlauben oder verweigern, wenn eine Anforderung oder Ressource die von Ihnen festgelegten Bedingungen erfüllt. Sie können den Zugriff für Prinzipale gewähren, denen Sie in anderen AWS-Konten-Konten vertrauen.

Authentifizierung mit Identitäten

Authentifizierung ist die Art, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffsportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuertem Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter Signieren von AWS-API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center-Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben

auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode empfiehlt es sich, menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, aufzufordern, den Verbund mit einem Identitätsanbieter zu verwenden, um auf AWS-Services mit temporären Anmeldeinformationen zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web Identity Provider, AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt werden, auf AWS-Services zugreift. Wenn Verbundidentitäten auf AWS-Konten zugreifen, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen im IAM Identity Center erstellen oder Sie können eine Verbindung mit einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie in allen AWS-Konten und Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie Rollen wechseln. Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff: Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center-Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen: Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff: Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward access sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur dann gestellt, wenn ein Service eine Anfrage erhält, die eine Interaktion mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle: Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Serviceverknüpfte Rolle: Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- Anwendungen in Amazon EC2: Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien,

die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Eine AWS KMS-Schlüsselrichtlinie ist eine ressourcenbasierte Richtlinie, die den Zugriff auf einen KMS-Schlüssel steuert. Jeder KMS-Schlüssel muss über eine Schlüsselrichtlinie verfügen. Sie können einen anderen Autorisierungsmechanismus verwenden, um den Zugriff auf den KMS-Schlüssel zu ermöglichen, jedoch nur, wenn die Schlüsselrichtlinie dies zulässt. (Sie können eine IAM-Richtlinie verwenden, um den Zugriff auf einen KMS-Schlüssel zu verweigern, auch wenn die Schlüsselrichtlinie dies nicht explizit zulässt.)

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource wie einen KMS-Schlüssel anfügen. Die ressourcenbasierte Richtlinie definiert die Aktionen, die ein bestimmter Prinzipal auf dieser Ressource durchführen kann und unter welchen Bedingungen. Sie geben die Ressource nicht in einer ressourcenbasierten Richtlinie an, sondern müssen einen Prinzipal angeben, z. B. Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services. Ressourcenbasierte Richtlinien sind Richtlinien, die sich in dem Service befinden, der die Ressource verwaltet. Sie können keine AWS-verwalteten Richtlinien von IAM, z. B. die `AWSKeyManagementServicePowerUser`-verwaltete Richtlinie, in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

AWS KMS unterstützt keine ACLs.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen:Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations-Benutzerhandbuch.
- Sitzungsrichtlinien:Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter Logik für die Richtlinienauswertung im IAM-Benutzerhandbuch.

AWS KMS-Ressourcen

In AWS KMS ist die primäre Ressource ein AWS KMS key. AWS KMS unterstützt auch einen Alias, eine unabhängige Ressource, die einen Anzeigenamen für einen KMS-Schlüssel bereitstellt. Bei manchen AWS KMS-Produktionen können Sie einen Alias verwenden, um einen KMS-Schlüssel zu identifizieren.

Jede Instance eines KMS-Schlüssels oder Aliases hat einen eindeutigen Amazon-Ressourcennamen (ARN) mit einem Standardformat. In AWS KMS-Ressourcen lautet der AWS-Servicename `kms`.

- AWS KMS key

  ARN-Format:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`AWS-Region`*`:`*`AWS-Konto ID`*`:key/`*`key ID`*

  Beispiel-ARN:

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

- Alias

  ARN-Format:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`AWS-Region`*`:`*`AWS-Konto ID`*`:alias/`*`alias name`*

  Beispiel-ARN:

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

AWS KMS bietet eine Reihe von API-Produktionen, über die Sie mit Ihren AWS KMS-Ressourcen arbeiten können. Weitere Hinweise zum Identifizieren von KMS-Schlüssel in den AWS Management Console- und AWS KMS-API-Produktionen finden Sie unter Schlüsselkennungen (KeyId). Eine Liste von AWS KMS-Produktionen finden Sie in der AWS Key Management Service-API-Referenz.

Wichtige Richtlinien in AWS KMS

Eine Schlüsselrichtlinie ist eine Ressourcenrichtlinie für eine. AWS KMS key Schlüsselrichtlinien sind die primäre Methode zur Zugriffssteuerung für KMS-Schlüssel. Jeder KMS-Schlüssel muss genau eine Schlüsselrichtlinie haben. Die Anweisungen im Schlüsselrichtliniendokument legen fest, wer über eine Berechtigung zur Verwendung des KMS-Schlüssels verfügt, und wie diese Verwendung erfolgen kann. Sie können den Zugriff auf den KMS-Schlüssel auch mithilfe von IAM-Richtlinien und Erteilungen steuern, jeder KMS-Schlüssel muss aber über ein Schlüsselrichtliniendokument verfügen.

Kein AWS Hauptbenutzer, auch nicht der Root-Benutzer oder der Ersteller des Schlüssels, hat Berechtigungen für einen KMS-Schlüssel, es sei denn, sie sind in einer Schlüsselrichtlinie, IAM-Richtlinie oder Grant ausdrücklich erlaubt und niemals verweigert.

Wenn die Schlüsselrichtlinie es nicht ausdrücklich erlaubt, können Sie keine IAM-Richtlinien verwenden, um den Zugriff auf einen KMS-Schlüssel zu erlauben. Ohne Berechtigung der Schlüsselrichtlinie haben IAM-Richtlinien, die Berechtigungen erlauben, keine Auswirkungen. (Sie können eine IAM-Richtlinie verwenden, um eine Berechtigung für einen KMS-Schlüssel ohne Berechtigung einer Schlüsselrichtlinie zu verweigern.) Die Standardschlüsselrichtlinie aktiviert IAM-Richtlinien. Um IAM-Richtlinien in Ihrer Schlüsselrichtlinie zu aktivieren, fügen Sie die unter Erlaubt den Zugriff auf das AWS-Konto und aktiviert IAM-Richtlinien beschriebene Richtlinienanweisung hinzu .

Im Gegensatz zu IAM-Richtlinien, die global sind, sind Schlüsselrichtlinien regional. Eine wichtige Richtlinie steuert den Zugriff nur auf einen KMS-Schlüssel in derselben Region. Es hat keine Auswirkungen auf KMS-Schlüssel in anderen Regionen.

Themen

- Erstellen einer Schlüsselrichtlinie
- Standardschlüsselrichtlinie
- Anzeigen einer Schlüsselrichtlinie
- Ändern einer Schlüsselrichtlinie
- Berechtigungen für AWS Dienste in wichtigen Richtlinien

Erstellen einer Schlüsselrichtlinie

Sie können wichtige Richtlinien in der AWS KMS Konsole mithilfe von AWS KMS API-Operationen wie, CreateKeyReplicateKeyPutKeyPolicy, und oder mithilfe einer AWS CloudFormation Vorlage erstellen und verwalten.

Wenn Sie einen KMS-Schlüssel in der AWS KMS Konsole erstellen, führt Sie die Konsole Schritt für Schritt durch die Erstellung einer Schlüsselrichtlinie, die auf der Standardschlüsselrichtlinie für die Konsole basiert. Wenn Sie die `CreateKey`- oder `ReplicateKey`-API verwenden, wenn Sie keine Schlüsselrichtlinie angeben, wenden diese APIs die Standardschlüsselrichtlinie für Schlüssel, die programmgesteuert erstellt wurden, an. Wenn Sie die `PutKeyPolicy`-API verwenden müssen Sie eine Schlüsselrichtlinie angeben.

Jedes Richtliniendokument kann eine oder mehrere Richtlinienanweisungen enthalten. Das folgende Beispiel zeigt ein gültiges Schlüsselrichtliniendokument mit einer Richtlinienanweisung.

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

Themen

- Schlüsselrichtlinienformat
- Elemente in einer Schlüsselrichtlinie
- Beispiel für eine Schlüsselrichtlinie

Schlüsselrichtlinienformat

Ein Schlüsselrichtliniendokument muss den folgenden Regeln entsprechen:

- Bis zu 32 Kilobytes (32 768 Bytes)
- Das `Sid`-Element in einer Schlüsselrichtlinienanweisung kann Leerzeichen enthalten. (Leerzeichen sind im `Sid`-Element eines IAM-Richtliniendokuments untersagt.)

Ein Schlüsselrichtliniendokument darf nur die folgenden Zeichen enthalten:

- Druckbare ASCII-Zeichen
- Die druckbaren Zeichen im zusätzlichen Zeichensatz Basic Latin und Latin-1 Supplement
- Die Sonderzeichen Tabulator (`\u0009`), Zeilenvorschub (`\u000A`) und Wagenrücklauf (`\u000D`)

Elemente in einer Schlüsselrichtlinie

Ein Schlüsselrichtliniendokument muss die folgenden Elemente besitzen:

Version

Gibt die Schlüsselrichtliniendokumentversion an. Wir empfehlen, die Version auf `2012-10-17` (neueste Version) einzustellen.

Statement

Fügt die Richtlinienanweisungen bei. Ein Schlüsselrichtliniendokument muss mindestens eine Anweisung enthalten.

Jede Schlüsselrichtlinienanweisung kann aus bis zu sechs Elementen bestehen. Die Elemente `Effect`, `Principal`, `Action`, und `Resource` sind erforderlich.

Sid

(Optional) Der Anweisungsbezeichner (`Sid`) ist eine beliebigen Zeichenfolge, die Sie zur Beschreibung der Anweisung verwenden können. Der `Sid` in einer Schlüsselrichtlinie kann Leerzeichen enthalten. (Sie können keine Leerzeichen in ein `Sid`-Element einer IAM-Richtlinie aufnehmen.)

Auswirkung

(Erforderlich) Gibt an, ob die Berechtigungen in der Richtlinienanweisung zugelassen oder verweigert werden. Gültige Werte sind `Allow` oder `Deny`. Wenn Sie den Zugriff auf einen KMS-Schlüssel nicht explizit erlauben, wird er implizit verweigert. Sie können den Zugriff auf einen KMS-Schlüssel auch explizit verweigern. So können Sie zum Beispiel sicherstellen, dass Benutzer nicht darauf zugreifen können, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.

Auftraggeber

(Erforderlich) Der Prinzipal ist die Identität, die die in der Richtlinienanweisung angegebenen Berechtigungen erhält. Sie können IAM-Benutzer AWS-Konten, IAM-Rollen und einige AWS Dienste als Principals in einer Schlüsselrichtlinie angeben. IAM-Benutzergruppen sind kein gültiger Prinzipal in irgendeinem Richtlinientyp.

Ein Sternchenwert, z. B. `"AWS": "*"`, steht für alle AWS -Identitäten in allen Konten.

> **Important**
>
> Setzen Sie den Prinzipal nicht auf ein Sternchen (*) in einer Schlüsselrichtlinienanweisung, die Berechtigungen erlaubt, es sei denn, Sie verwenden Bedingungen, um die Schlüsselrichtlinie einzuschränken. Ein Sternchen gibt jede Identität in jeder AWS-Konto Berechtigung zur Verwendung des KMS-Schlüssels an, sofern dies nicht in einer anderen Richtlinienanweisung ausdrücklich verweigert wird. Benutzer in anderen Ländern AWS-Konten können Ihren KMS-Schlüssel immer dann verwenden, wenn sie über entsprechende Berechtigungen in ihrem eigenen Konto verfügen.

> **Note**
>
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Wenn der Prinzipal in einer wichtigen Richtlinienanweisung ein AWS-Konto -Prinzipal ist, das als `arn:aws:iam::111122223333:root"` ausgedrückt wird, erteilt die Richtlinienanweisung keinem IAM-Prinzipal die Berechtigung. Stattdessen erteilt es die AWS-Konto Erlaubnis, IAM-Richtlinien zu verwenden, um die in der Schlüsselrichtlinie angegebenen Berechtigungen zu delegieren. (Ein Prinzipal im `arn:aws:iam::111122223333:root"`-Format repräsentiert nicht die AWS -Stammbenutzer des Kontos, trotz der Verwendung von „root“ in der Kontokennung. Der Kontoprinzipal repräsentiert jedoch das Konto und seine Administratoren, einschließlich des Account-Root-Benutzers.)

Handelt es sich bei dem Prinzipal um einen anderen Prinzipal AWS-Konto oder dessen Prinzipale, sind die Berechtigungen nur wirksam, wenn das Konto in der Region mit dem KMS-Schlüssel und der Schlüsselrichtlinie aktiviert ist. Informationen zu Regionen, die standardmäßig nicht aktiviert sind („Opt-In-Regionen“), finden Sie unter Verwalten von AWS-Regionen in Allgemeine AWS-Referenz.

Um einem anderen AWS-Konto oder seinen Prinzipalen die Verwendung eines KMS-Schlüssels zu ermöglichen, müssen Sie in einer Schlüsselrichtlinie und in einer IAM-Richtlinie im anderen Konto die entsprechenden Berechtigungen erteilen. Details hierzu finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

Aktion

(Erforderlich) Geben Sie die zugelassenen oder verweigerten API-Operationen an. Die `kms:Encrypt` Aktion entspricht beispielsweise dem Vorgang AWS KMS Verschlüsseln. Sie können mehr als eine Aktion in einer Richtlinienanweisung auflisten. Weitere Informationen finden Sie unter Berechtigungsreferenz.

Ressource

(Erforderlich) In einer Schlüsselrichtlinie ist der Wert des Ressourcen-Elements `"*"`, was "dieser KMS-Schlüssel" bedeutet. Das Sternchen (`"*"`) identifiziert den KMS-Schlüssel, an den die Schlüsselrichtlinie angefügt ist.

> **Note**
>
> Wenn das erforderliche `Resource`-Element in einer Schlüsselrichtlinienanweisung fehlt, hat die Richtlinienanweisung keine Wirkung. Eine Schlüsselrichtlinie ohne `Resource`-Element gilt für keinen KMS-Schlüssel.

> Wenn das `Resource` Element einer wichtigen Richtlinienanweisung fehlt, meldet die AWS KMS Konsole korrekt einen Fehler, aber die PutKeyPolicyAPIs CreateKeyund sind erfolgreich, obwohl die Richtlinienanweisung unwirksam ist.

Bedingung

(Optional) Bedingungen geben an, welche Anforderungen erfüllt werden müssen, damit eine Schlüsselrichtlinie wirksam wird. AWS Kann unter bestimmten Bedingungen den Kontext einer API-Anfrage auswerten, um festzustellen, ob die Grundsatzerklärung zutrifft oder nicht.

Um Bedingungen anzugeben, verwenden Sie vordefinierte Bedingungsschlüssel. AWS KMS unterstützt AWS globale Bedingungsschlüssel und AWS KMS Bedingungsschlüssel. Zur Unterstützung der attributebasierten Zugriffskontrolle (ABAC) AWS KMS stellt es Bedingungsschlüssel bereit, die den Zugriff auf einen KMS-Schlüssel auf der Grundlage von Tags und Aliasen steuern. Details hierzu finden Sie unter ABAC für AWS KMS.

Das Format für eine Bedingung lautet:

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

wie beispielsweise:

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

Weitere Informationen zur AWS Richtliniensyntax finden Sie unter AWS IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Beispiel für eine Schlüsselrichtlinie

Das folgende Beispiel zeigt eine vollständige Schlüsselrichtlinie für einen KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können es als Nachschlagewerk verwenden, wenn Sie sich mit den wichtigsten Konzepten in diesem Kapitel befassen. Diese Schlüsselrichtlinie vereint die Beispiel-Richtlinienanweisungen des vorherigen Abschnitts Standardschlüsselrichtlinie in einer einzigen Schlüsselrichtlinie, wodurch Folgendes erreicht wird:

- Ermöglicht im Beispiel AWS-Konto 111122223333 vollen Zugriff auf den KMS-Schlüssel. Damit können das Konto und seine Administratoren, einschließlich des Root-Benutzers des Kontos (in Notfällen), IAM-Richtlinien verwenden, um den Zugriff auf den KMS-Schlüssel zu gestatten.

- Erlaubt der `ExampleAdminRole`-IAM-Rolle den KMS-Schlüssel zu verwalten.
- Erlaubt der `ExampleUserRole`-IAM-Rolle, den KMS-Schlüssel zu verwenden.

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

Standardschlüsselrichtlinie

Beim Erstellen eines KMS-Schlüssels können Sie die Schlüsselrichtlinie für den neuen KMS-Schlüssel angeben. Wenn Sie keinen angeben, AWS KMS erstellt er einen für Sie. Die verwendete Standardschlüsselrichtlinie unterscheidet sich je nachdem, ob Sie den Schlüssel in der AWS KMS Konsole erstellen oder die AWS KMS API verwenden. AWS KMS

Die Standard-Schlüsselrichtlinie beim programmgesteuerten Erstellen eines KMS-Schlüssels

Wenn Sie einen KMS-Schlüssel programmgesteuert mit der AWS KMS API erstellen (auch mithilfe der AWS SDKs AWS Command Line Interfaceoder AWS Tools for PowerShell) und Sie keine Schlüsselrichtlinie angeben, wird eine sehr einfache Standardschlüsselrichtlinie AWS KMS angewendet. Diese Standardschlüsselrichtlinie enthält eine Richtlinienanweisung, die demjenigen, der den KMS-Schlüssel besitzt AWS-Konto , die Erlaubnis erteilt, IAM-Richtlinien zu verwenden, um Zugriff auf alle AWS KMS Operationen mit dem KMS-Schlüssel zu gewähren. Weitere Informationen zu dieser Richtlinienanweisung finden Sie unter Erlaubt den Zugriff auf das AWS-Konto und aktiviert IAM-Richtlinien.

Standardschlüsselrichtlinie beim Erstellen eines KMS-Schlüssels mit AWS Management Console

Wenn Sie einen KMS-Schlüssel mit dem erstellen AWS Management Console, beginnt die Schlüsselrichtlinie mit der Richtlinienanweisung, die den Zugriff auf die IAM-Richtlinien ermöglicht AWS-Konto und diese aktiviert. Die Konsole fügt dann eine wichtige Administratoranweisung, eine Schlüsselbenutzeranweisung und (bei den meisten Schlüsseltypen) eine Anweisung hinzu, die es Prinzipalen ermöglicht, den KMS-Schlüssel mit anderen AWS Diensten zu verwenden. Sie können die Funktionen der AWS KMS Konsole verwenden, um die IAM-Benutzer, die IAM-Rollen und die Personen, die Schlüsseladministratoren sind, und AWS-Konten diejenigen, die Hauptbenutzer sind (oder beides), anzugeben.

Berechtigungen

- Erlaubt den Zugriff auf das AWS-Konto und aktiviert IAM-Richtlinien
- Erlaubt Schlüsseladministratoren die Verwaltung des KMS-Schlüssels
- Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssels
  - Erlaubt Schlüsselbenutzern die Verwendung eines KMS-Schlüssels für kryptografische Operationen
  - Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssel mit AWS -Services

Erlaubt den Zugriff auf das AWS-Konto und aktiviert IAM-Richtlinien

Die folgende standardmäßige Schlüsselrichtlinienanweisung ist kritisch.

- Sie gewährt demjenigen AWS-Konto , der den KMS-Schlüssel besitzt, vollen Zugriff auf den KMS-Schlüssel.

  Im Gegensatz zu anderen AWS Ressourcenrichtlinien erteilt eine AWS KMS Schlüsselrichtlinie nicht automatisch Berechtigungen für das Konto oder eine seiner Identitäten. Um

Kontoadministratoren die Berechtigung zu erteilen, muss die Schlüsselrichtlinie eine explizite Erklärung enthalten, die diese Berechtigung wie diese bereitstellt.

- Zusätzlich zur Schlüsselrichtlinie kann das Konto IAM-Richtlinien verwenden, um den Zugriff auf den KMS-Schlüssel zu gestatten.

  Ohne diese Berechtigung sind IAM-Richtlinien, die den Zugriff auf den Schlüssel ermöglichen, wirkungslos, obwohl IAM-Richtlinien, die den Zugriff auf den Schlüssel verweigern, weiterhin wirksam sind.
- Es reduziert das Risiko, dass der Schlüssel unüberschaubar wird, indem es den Kontoadministratoren, einschließlich dem Account-Root-Benutzer, die nicht gelöscht werden kann, die Zugriffsberechtigung erteilt.

Die folgende wichtige Richtlinienanweisung ist die gesamte Standardschlüsselrichtlinie für KMS-Schlüssel, die programmgesteuert erstellt wurden. Dies ist die erste Richtlinienanweisung in der Standard-Schlüsselrichtlinie für KMS-Schlüssel, die in der AWS KMS Konsole erstellt wurde.

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

Erlaubt es IAM-Richtlinien, den Zugriff auf den KMS-Schlüssel zu steuern.

Die oben abgebildete Grundsatzerklärung gibt AWS-Konto demjenigen, der den Schlüssel besitzt, die Erlaubnis, IAM-Richtlinien sowie wichtige Richtlinien zu verwenden, um alle Aktionen (`kms:*`) für den KMS-Schlüssel zuzulassen.

Der Grundsatz in dieser wichtigen Richtlinienanweisung ist der Kontoprinzipal, der durch einen ARN in diesem Format repräsentiert wird: `arn:aws:iam::account-id:root`. Der Kontoprinzipal repräsentiert das AWS Konto und seine Administratoren.

Wenn der Prinzipal in einer Schlüsselrichtlinienausweisung der Kontoprinzipal ist, erteilt die Richtlinienerklärung keinen IAM-Identitäten die Berechtigung, den KMS-Schlüssel zu verwenden. Stattdessen erlaubt es dem Konto, IAM-Richtlinien zu verwenden, um die in der

Richtlinienanweisung angegebenen Berechtigungen zu delegieren. Diese standardmäßige Schlüsselrichtlinienausweisung ermöglicht es dem Konto, IAM-Richtlinien zu verwenden, um Berechtigungen für alle Aktionen zu delegieren (`kms:*`) auf dem KMS-Schlüssel.

verringert das Risiko, dass der KMS-Schlüssel nicht mehr verwaltet werden kann.

Im Gegensatz zu anderen AWS Ressourcenrichtlinien erteilt eine AWS KMS Schlüsselrichtlinie dem Konto oder einem seiner Hauptbenutzer nicht automatisch Berechtigungen. Um jedem Auftraggeber die Berechtigung zu erteilen, einschließlich der-Kontoauftraggeber verwenden, müssen Sie eine Schlüsselrichtlinienanweisung verwenden, die die Berechtigung explizit bereitstellt. Sie müssen dem Kontoprinzipal oder einem Prinzipal keinen Zugriff auf den KMS-Schlüssel gewähren. Wenn Sie jedoch Zugriff auf den Kontoprinzipal gewähren, können Sie verhindern, dass der Schlüssel nicht überschaubar wird.

Angenommen, Sie erstellen eine Schlüsselrichtlinie, die nur einem Benutzer Zugriff auf den KMS-Schlüssel gewährt. Wenn Sie diesen Benutzer dann löschen, wird der Schlüssel unüberschaubar und Sie müssen den AWS -Support kontaktieren, um wieder Zugriff auf den KMS-Schlüssel zu erhalten.

Mit der oben abgebildeten Grundsatzerklärung wird die Kontrolle über den Schlüssel zum Kontoprinzipal erteilt, der das Konto AWS-Konto und seine Administratoren, einschließlich des Root-Benutzers, repräsentiert. Der Account-Root-Benutzer ist der einzige Prinzipal, der nur gelöscht werden kann, wenn Sie den AWS-Konto löschen. Die Best Practices von IAM raten davon ab, im Namen des Account-Root-Benutzers zu handeln, außer im Notfall. Möglicherweise müssen Sie jedoch als Account-Root-Benutzer fungieren, wenn Sie alle anderen Benutzer und Rollen mit Zugriff auf den KMS-Schlüssel löschen.

Erlaubt Schlüsseladministratoren die Verwaltung des KMS-Schlüssels

Die von der Konsole erstellte Standard-Schlüsselrichtlinie erlaubt es Ihnen, IAM-Benutzer und -Rollen im Konto auszuwählen und diese zu Schlüsseladministratoren zu machen. Diese Anweisung wird Schlüsseladministratoren-Anweisung genannt. Schlüsseladministratoren sind dazu berechtigt, den KMS-Schlüssel zu verwalten, sie verfügen jedoch nicht über die Berechtigungen, den KMS-Schlüssel in kryptografischen Operationen zu verwenden. Sie können IAM-Benutzer und -Rollen zu der Liste der Schlüsseladministratoren hinzufügen, wenn Sie den KMS-Schlüssel in der Standard- oder Richtlinienansicht erstellen.

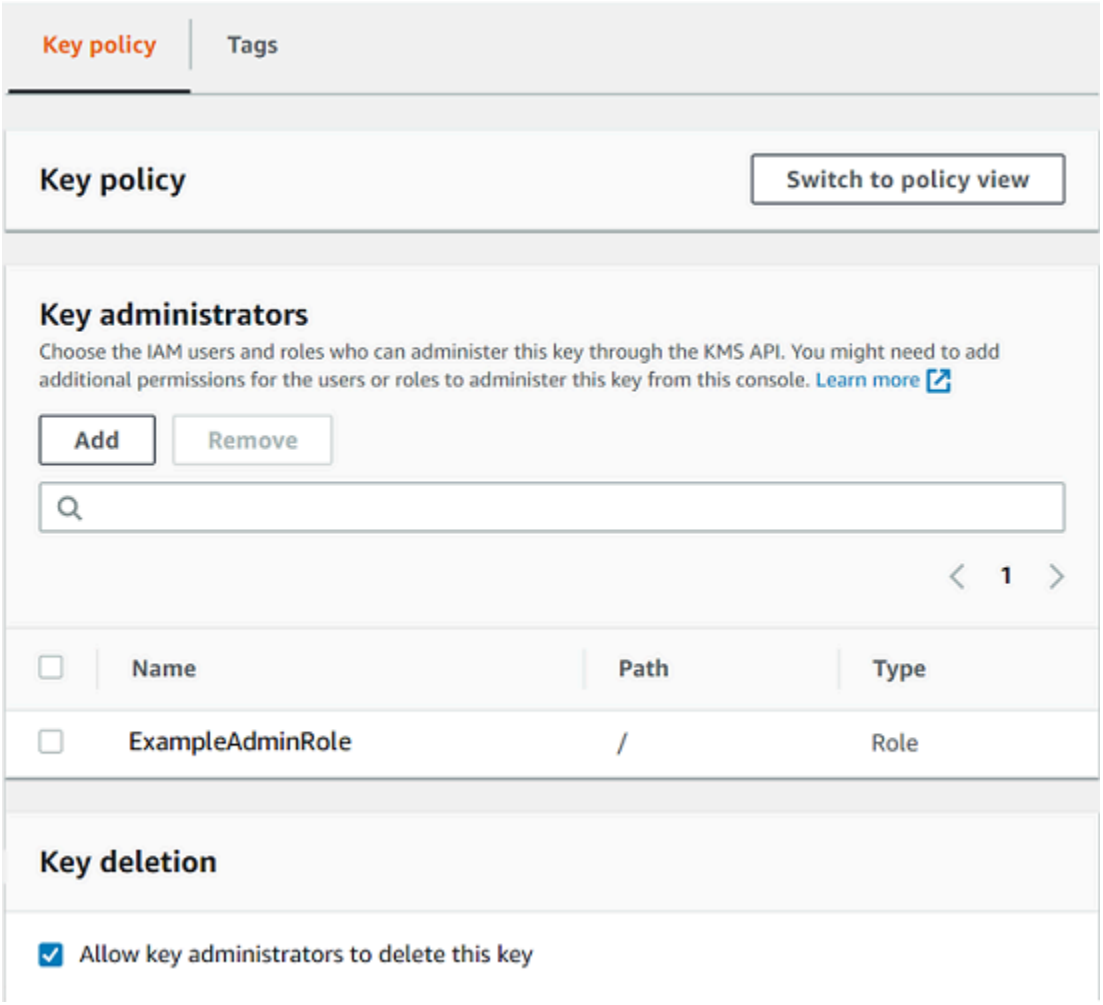
Warning

Da Schlüsseladministratoren berechtigt sind, die Schlüsselrichtlinie zu ändern und Zuweisungen zu erstellen, können sie sich selbst und anderen AWS KMS Berechtigungen gewähren, die in dieser Richtlinie nicht spezifiziert sind.
Prinzipale, die über die Berechtigung zum Verwalten von Tags und Aliasen verfügen, können auch den Zugriff auf einen KMS-Schlüssel steuern. Details hierzu finden Sie unter ABAC für AWS KMS.

Note

Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Das folgende Beispiel zeigt die Schlüsseladministratoren-Anweisung in der Standardansicht der AWS KMS -Konsole.



Nachfolgend sehen Sie ein Beispiel für eine Schlüsseladministratoren-Anweisung in der Richtlinienansicht der AWS KMS -Konsole. Diese Schlüsseladministratoren-Anweisung bezieht sich auf einen KMS-Schlüssel mit symmetrischer Verschlüsselung für eine Region.

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

Die standardmäßige Schlüsseladministratoren-Anweisung für den geläufigsten KMS-Schlüssel, ein KMS-Schlüssel mit symmetrischer Verschlüsselung für eine Region, erteilt die folgenden Berechtigungen. Ausführliche Informationen zu den einzelnen Berechtigungen finden Sie unter AWS KMS Berechtigungen.

Wenn Sie die AWS KMS Konsole verwenden, um einen KMS-Schlüssel zu erstellen, fügt die Konsole die von Ihnen angegebenen Benutzer und Rollen dem `Principal` Element in der Anweisung für wichtige Administratoren hinzu.

Viele dieser Berechtigungen enthalten das Platzhalterzeichen (*), sodass alle Berechtigungen zugelassen werden, die mit dem angegebenen Verb beginnen. Wenn neue API-Operationen AWS KMS hinzugefügt werden, dürfen Schlüsseladministratoren diese daher automatisch verwenden. Sie müssen Ihre Schlüsselrichtlinien nicht aktualisieren, um die neuen Operationen einzubeziehen. Wenn Sie Ihre Schlüsseladministratoren lieber auf bestimmte API-Operationen beschränken möchten, können Sie Ihre Schlüsselrichtlinie ändern.

`kms:Create*`

Erlaubt `kms:CreateAlias` und `kms:CreateGrant`. (Die `kms:CreateKey`-Berechtigung ist nur in einer IAM-Richtlinie gültig.)

`kms:Describe*`

Erlaubt `kms:DescribeKey`. Die Berechtigung `kms:DescribeKey` ist erforderlich, um die Seite mit den Schlüsseldetails für einen KMS-Schlüssel in der AWS Management Console anzuzeigen.

`kms:Enable*`

Erlaubt `kms:EnableKey`. Für KMS-Schlüssel mit symmetrischer Verschlüsselung erlaubt sie auch `kms:EnableKeyRotation`.

`kms:List*`

Erlaubt `kms:ListGrants`, `kms:ListKeyPolicies` und `kms:ListResourceTags`. (Die Berechtigungen `kms:ListAliases` und `kms:ListKeys`, die zum Anzeigen von KMS-Schlüsseln in der AWS Management Console erforderlich sind, sind nur in IAM-Richtlinien gültig.)

`kms:Put*`

Erlaubt `kms:PutKeyPolicy`. Diese Berechtigung erlaubt Schlüsseladministratoren, die Schlüsselrichtlinie für diesen KMS-Schlüssel zu ändern.

`kms:Update*`

Erlaubt `kms:UpdateAlias` und `kms:UpdateKeyDescription`. Bei multiregionalen Schlüsseln erlaubt sie `kms:UpdatePrimaryRegion` für diesen KMS-Schlüssel.

`kms:Revoke*`

Erlaubt `kms:RevokeGrant`, wodurch Schlüsseladministratoren eine Erteilung löschen können, auch wenn sie kein ausscheidender Prinzipal in der Erteilung sind.

`kms:Disable*`

Erlaubt `kms:DisableKey`. Für KMS-Schlüssel mit symmetrischer Verschlüsselung erlaubt sie auch `kms:DisableKeyRotation`.

`kms:Get*`

Erlaubt `kms:GetKeyPolicy` und `kms:GetKeyRotationStatus`. Für KMS-Schlüssel mit importiertem Schlüsselmaterial erlaubt sie `kms:GetParametersForImport`. Für asymmetrische KMS-Schlüssel erlaubt sie `kms:GetPublicKey`. Die Berechtigung `kms:GetKeyPolicy` ist erforderlich, um die Schlüsselrichtlinie eines KMS-Schlüssels in der AWS Management Console anzuzeigen.

`kms:Delete*`

Erlaubt `kms:DeleteAlias`. Für Schlüssel mit importiertem Schlüsselmaterial erlaubt sie `kms:DeleteImportedKeyMaterial`. Die Berechtigung `kms:Delete*` erlaubt Schlüsseladministratoren nicht das Löschen des KMS-Schlüssels (`ScheduleKeyDeletion`).

`kms:TagResource`

Erlaubt `kms:TagResource`, sodass Schlüsseladministratoren Tags zum KMS-Schlüssel hinzufügen können. Da Tags auch verwendet werden können, um den Zugriff auf den KMS-

Schlüssel zu steuern, können Administratoren mit dieser Berechtigung den Zugriff auf den KMS-Schlüssel gewähren oder verweigern. Details hierzu finden Sie unter ABAC für AWS KMS.

`kms:UntagResource`

Erlaubt `kms:UntagResource`, sodass Schlüsseladministratoren Tags aus dem KMS-Schlüssel löschen können. Da Tags verwendet werden können, um den Zugriff auf den Schlüssel zu steuern, können Administratoren mit dieser Berechtigung den Zugriff auf den KMS-Schlüssel gewähren oder verweigern. Details hierzu finden Sie unter ABAC für AWS KMS.

`kms:ScheduleKeyDeletion`

Erlaubt `kms:ScheduleKeyDeletion`, sodass Schlüsseladministratoren diesen KMS-Schlüssel löschen können. Um diese Berechtigung zu löschen, deaktivieren Sie die Option Allow key administrators to delete this key (Schlüsseladministratoren das Löschen dieses Schlüssels erlauben).

`kms:CancelKeyDeletion`

Erlaubt `kms:CancelKeyDeletion`, sodass Schlüsseladministratoren das Löschen dieses KMS-Schlüssels abbrechen können. Um diese Berechtigung zu löschen, deaktivieren Sie die Option Allow key administrators to delete this key (Schlüsseladministratoren das Löschen dieses Schlüssels erlauben).

AWS KMS fügt der Standardanweisung für Schlüsseladministratoren die folgenden Berechtigungen hinzu, wenn Sie Schlüssel für spezielle Zwecke erstellen.

`kms:ImportKeyMaterial`

Die Berechtigung `kms:ImportKeyMaterial` erlaubt es Schlüsseladministratoren, Schlüsselmaterial in den KMS-Schlüssel zu importieren. Diese Berechtigung ist nur in der Schlüsselrichtlinie enthalten, wenn Sie einen KMS-Schlüssel ohne Schlüsselmaterial erstellen.

`kms:ReplicateKey`

Diese `kms:ReplicateKey`Berechtigung ermöglicht es Schlüsseladministratoren, ein Replikat eines Primärschlüssels für mehrere Regionen in einer anderen Region zu erstellen. AWS Diese Berechtigung ist nur dann in der Schlüsselrichtlinie enthalten, wenn Sie einen multiregionalen Primär- oder Replikatschlüssel erstellen.

`kms:UpdatePrimaryRegion`

Die Berechtigung `kms:UpdatePrimaryRegion` erlaubt es Schlüsseladministratoren, einen multiregionalen Replikatschlüssel zu einem multiregionalen Primärschlüssel zu ändern. Diese Berechtigung ist nur dann in der Schlüsselrichtlinie enthalten, wenn Sie einen multiregionalen Primär- oder Replikatschlüssel erstellen.

Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssels

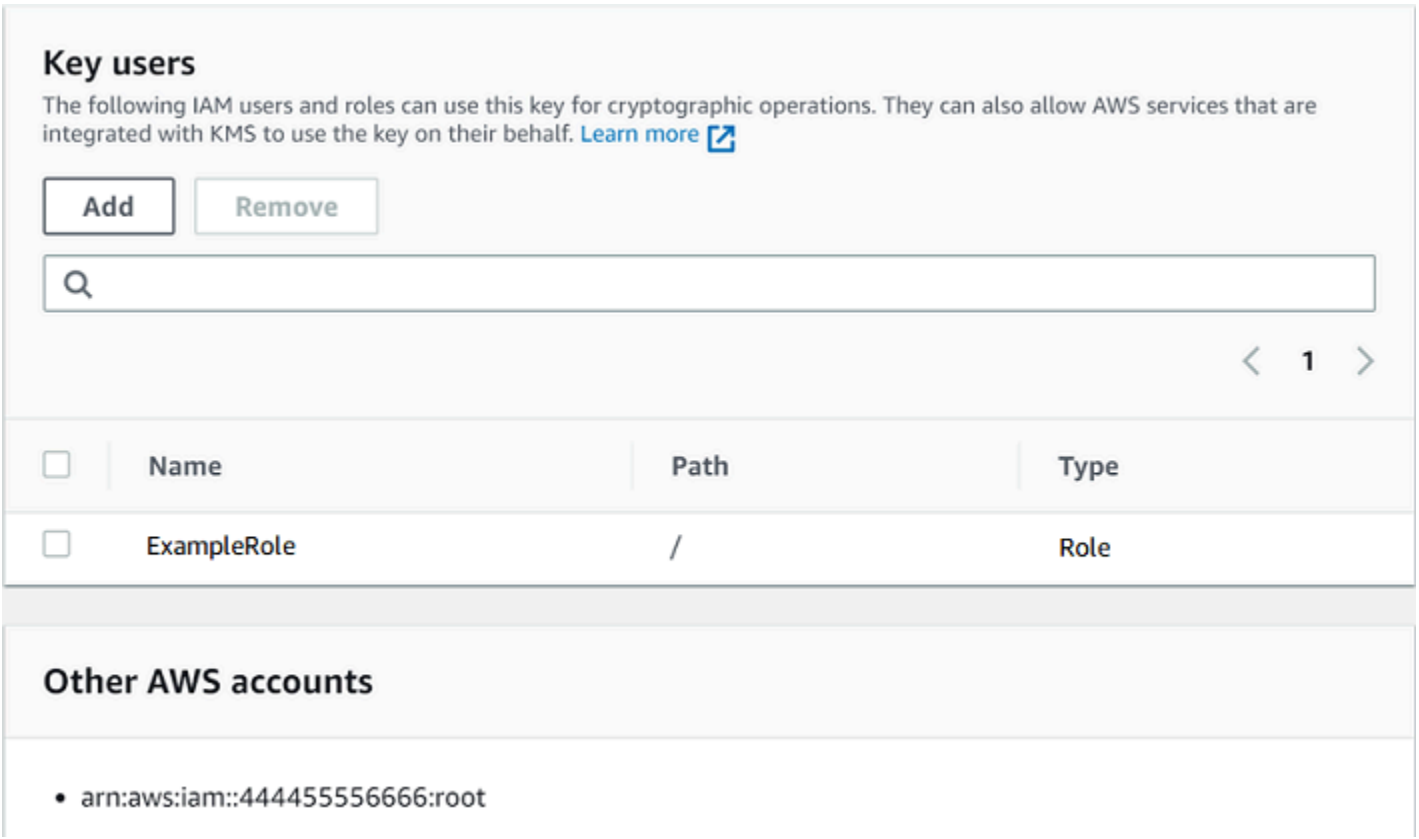
Die Standardschlüsselrichtlinie, die die Konsole für KMS-Schlüssel erstellt, ermöglicht es Ihnen, IAM-Benutzer und IAM-Rollen innerhalb des Kontos sowie externe AWS-Konten Benutzer auszuwählen und sie zu Schlüsselbenutzern zu machen.

Die Konsole fügt der Schlüsselrichtlinie für Schlüsselbenutzer zwei Richtlinienanweisungen hinzu.

- Direkte Verwendung des KMS-Schlüssels – Die erste Schlüsselrichtlinienanweisung gibt Schlüsselbenutzern die Berechtigung, den KMS-Schlüssel direkt für alle unterstützten kryptografischen Operationen für diesen KMS-Schlüsseltyp zu verwenden.
- Verwenden Sie den KMS-Schlüssel mit AWS Diensten — Die zweite Richtlinienanweisung gibt Schlüsselbenutzern die Erlaubnis, AWS Diensten, die in integriert sind, die Verwendung des KMS-Schlüssels in ihrem Namen zu gestatten, um Ressourcen wie Amazon S3 S3-Buckets und Amazon DynamoDB-Tabellen zu schützen. AWS KMS

Sie können IAM-Benutzer, IAM-Rollen und andere AWS-Konten zur Liste der Hauptbenutzer hinzufügen, wenn Sie den KMS-Schlüssel erstellen. Sie können auch die Liste der Schlüsselrichtlinien in der Standardansicht der Konsole wie in der folgenden Abbildung dargestellt bearbeiten. Die Standardansicht für Schlüsselrichtlinien finden Sie auf der Seite mit den Schlüsseldetails. Weitere Informationen darüber, wie Benutzern in anderen AWS-Konten Ländern die Verwendung des KMS-Schlüssels gestattet wird, finden Sie unter. Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben

> **Note**
>
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.



Die Standard-Schlüsselbenutzer-Anweisung für einen symmetrischen Schlüssel für eine Region gewährt die folgenden Berechtigungen. Ausführliche Informationen zu den einzelnen Berechtigungen finden Sie unter AWS KMS Berechtigungen.

Wenn Sie die AWS KMS Konsole verwenden, um einen KMS-Schlüssel zu erstellen, fügt die Konsole die Benutzer und Rollen, die Sie angeben, dem `Principal` Element in den einzelnen Schlüsselbenutzeranweisungen hinzu.

```
{
  "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}}
}
```

Erlaubt Schlüsselbenutzern die Verwendung eines KMS-Schlüssels für kryptografische Operationen

Schlüsselbenutzer haben die Berechtigung, den KMS-Schlüssel direkt in allen kryptografischen Operationen zu verwenden, die vom KMS-Schlüssel unterstützt werden. Sie können den DescribeKeyVorgang auch verwenden, um detaillierte Informationen über den KMS-Schlüssel in der AWS KMS Konsole oder mithilfe der AWS KMS API-Operationen abzurufen.

Standardmäßig fügt die AWS KMS Konsole der Standard-Schlüsselrichtlinie wichtige Benutzeranweisungen wie die in den folgenden Beispielen hinzu. Da sie unterschiedliche API-Operationen unterstützen, weichen die jeweiligen Aktionen in den Richtlinienanweisungen für KMS-Schlüssel mit symmetrischer Verschlüsselung, HMAC-KMS-Schlüssel, asymmetrische KMS-Schlüssel für die Verschlüsselung öffentlicher Schlüssel und asymmetrische KMS-Schlüssel für Signatur und Verifizierung geringfügig voneinander ab.

KMS-Schlüssel mit symmetrischer Verschlüsselung

Die Konsole fügt der Schlüsselrichtlinie für KMS-Schlüssel mit symmetrischer Verschlüsselung die folgende Anweisung hinzu.

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

HMAC-KMS-Schlüssel

Die Konsole fügt der Schlüsselrichtlinie für HMAC-KMS-Schlüssel die folgende Anweisung hinzu.

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

Asymmetrische KMS-Schlüssel für die Verschlüsselung öffentlicher Schlüssel

Die Konsole fügt die folgende Anweisung zur Schlüsselrichtlinie für asymmetrische KMS-Schlüssels mit der Schlüsselnutzung Encrypt and decrypt (Verschlüsseln und Entschlüsseln) hinzu.

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

Asymmetrische KMS-Schlüssel für Signatur und Verifizierung

Die Konsole fügt die folgende Anweisung zur Schlüsselrichtlinie für asymmetrische KMS-Schlüssels mit der Schlüsselnutzung Sign and verify (Signieren und Überprüfen) hinzu.

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

Die Aktionen in diesen Anweisungen geben den Schlüsselbenutzern die folgenden Berechtigungen.

`kms:Encrypt`

Erlaubt es Schlüsselbenutzern, Daten mit diesem KMS-Schlüssel zu verschlüsseln.

`kms:Decrypt`

Erlaubt es Schlüsselbenutzern, Daten mit diesem KMS-Schlüssel zu entschlüsseln.

`kms:DescribeKey`

Erlaubt Schlüsselbenutzern das Abrufen detaillierter Information über diesen KMS-Schlüssel, einschließlich der dazugehörigen IDs, des Erstellungsdatums und des Schlüsselstatus. Außerdem können die Hauptbenutzer Details zum KMS-Schlüssel in der AWS KMS Konsole anzeigen.

`kms:GenerateDataKey*`

Erlaubt Schlüsselbenutzern, einen symmetrischen Datenschlüssel oder ein asymmetrisches Datenschlüsselpaar für clientseitige kryptografische Operationen anzufordern. Die Konsole verwendet das Platzhalterzeichen *, um die Erlaubnis für die folgenden API-Operationen darzustellen: GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair, und GenerateDataKeyPairWithoutPlaintext. Diese Berechtigungen gelten nur für die symmetrischen KMS-Schlüssel, die die Datenschlüssel verschlüsseln.

km: GenerateMac

Ermöglicht es Schlüsselbenutzern, einen HMAC-KMS-Schlüssel zu verwenden, um ein HMAC-Tag zu generieren.

km: GetPublicKey

Erlaubt Schlüsselbenutzern das Herunterladen des öffentlichen Schlüssels des asymmetrischen KMS-Schlüssels. Parteien, mit denen Sie diesen öffentlichen Schlüssel teilen, können Daten außerhalb von AWS KMS verschlüsseln. Diese Chiffretexte können jedoch nur durch Aufrufen der Produktion Decrypt in AWS KMS entschlüsselt werden.

km: * ReEncrypt

Erlaubt es Schlüsselbenutzern, Daten erneut zu verschlüsseln, die ursprünglich mit diesem KMS-Schlüssel verschlüsselt wurden, oder diesen KMS-Schlüssel zu verwenden, um zuvor verschlüsselte Daten erneut zu verschlüsseln. Der ReEncryptVorgang erfordert Zugriff sowohl auf Quell- als auch auf Ziel-KMS-Schlüssel. Um dies zu erreichen, können Sie die `kms:ReEncryptFrom`-Berechtigung für den Quell-KMS-Schlüssel und die `kms:ReEncryptTo`-Berechtigung für den Ziel-KMS-Schlüssel erlauben. Der Einfachheit halber erlaubt die Konsole jedoch `kms:ReEncrypt*` (mit dem Platzhalterzeichen `*`) für beide KMS-Schlüssel.

kms:Sign

Erlaubt es Schlüsselbenutzern, Nachrichten mit diesem KMS-Schlüssel zu signieren.

kms:Verify

Erlaubt es Schlüsselbenutzern, Signaturen mit diesem KMS-Schlüssel zu überprüfen.

km: VerifyMac

Ermöglicht es Schlüsselbenutzern, einen HMAC-KMS-Schlüssel zu verwenden, um ein HMAC-Tag zu verifizieren.

Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssel mit AWS -Services

Die standardmäßige Schlüsselrichtlinie in der Konsole gewährt Schlüsselbenutzern außerdem die Zugriffsberechtigungen, die sie zum Schutz ihrer Daten in AWS Diensten benötigen, die Grants verwenden. AWS Dienste verwenden häufig Zuschüsse, um spezifische und eingeschränkte Berechtigungen zur Verwendung eines KMS-Schlüssels zu erhalten.

Diese wichtige Richtlinienerklärung ermöglicht es dem Hauptbenutzer, Berechtigungen für den KMS-Schlüssel zu erstellen, einzusehen und zu widerrufen, aber nur, wenn die Anfrage für den Gewährungsvorgang von einem AWS Dienst stammt, in den der Dienst integriert ist AWS KMS. Die GrantIsFor AWSResource Richtlinienbedingung kms: erlaubt es dem Benutzer nicht, diese Grant-Operationen direkt aufzurufen. Wenn der Schlüsselbenutzer dies zulässt, kann ein AWS Dienst im Namen des Benutzers eine Gewährung einrichten, die es dem Dienst ermöglicht, den KMS-Schlüssel zum Schutz der Benutzerdaten zu verwenden.

Schlüsselbenutzer benötigen diese Erteilungs-Berechtigungen, um ihren KMS-Schlüssel mit integrierten Services zu verwenden, aber diese Berechtigungen sind nicht ausreichend. Schlüsselbenutzer benötigen außerdem die Berechtigung, die integrierten Services zu nutzen. Einzelheiten darüber, wie Sie Benutzern Zugriff auf einen AWS Dienst gewähren, der integriert werden kann AWS KMS, finden Sie in der Dokumentation zum integrierten Dienst.

```
{
  "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}}
}
```

Schlüsselbenutzer können diese Berechtigungen beispielsweise auf folgende Weise für den KMS-Schlüssel verwenden.

- Verwenden Sie diesen KMS-Schlüssel mit Amazon Elastic Block Store (Amazon EBS) und Amazon Elastic Compute Cloud (Amazon EC2), um ein verschlüsseltes EBS-Volume an eine EC2-Instance anzufügen. Der Schlüsselbenutzer erteilt Amazon EC2 implizit die Berechtigung, den KMS-

Schlüssel zu verwenden, um das verschlüsselte Volume an die Instance anzufügen. Weitere Informationen finden Sie unter Wie Amazon Elastic Block Store (Amazon EBS) AWS KMS nutzt..

- Verwenden Sie diesen KMS-Schlüssel zusammen mit Amazon Redshift, um einen verschlüsselten Cluster zu starten. Der Schlüsselbenutzer erteilt Amazon Redshift implizit die Berechtigung, den KMS-Schlüssel zu verwenden, um den verschlüsselten Cluster zu starten und verschlüsselte Snapshots zu erstellen. Weitere Informationen finden Sie unter Wie Amazon Redshift AWS KMS nutzt.
- Verwenden Sie diesen KMS-Schlüssel mit anderen AWS -Diensten, die in AWS KMS integriert sind und Berechtigungen zum Erstellen, Verwalten oder Verwenden verschlüsselter Ressourcen mit diesen Diensten verwenden.

Die Standard-Berechtigung ermöglicht es Schlüsselbenutzern, alle integrierten Services zu verwenden, die Zugriffserteilungen nutzen. Sie können jedoch eine benutzerdefinierte Schlüsselrichtlinie erstellen, die die Berechtigungen auf bestimmte AWS Dienste beschränkt. Weitere Informationen erhalten Sie unter km: ViaService Bedingungsschlüssel.

Anzeigen einer Schlüsselrichtlinie

Sie können die Schlüsselrichtlinie für einen vom AWS KMS Kunden verwalteten Schlüssel oder einen Von AWS verwalteter Schlüssel in Ihrem Konto anzeigen, indem Sie die - AWS Management Console oder die -GetKeyPolicyOperation in der AWS KMS-API verwenden. Sie können diese Methoden nicht verwenden, um die Schlüsselrichtlinie eines KMS-Schlüssels in einem anderen AWS-Konto anzuzeigen.

Weitere Informationen zu AWS KMS-Schlüsselrichtlinien finden Sie unter Wichtige Richtlinien in AWS KMS. Informationen zum Bestimmen, welche Benutzer und Rollen Zugriff auf ein KMS-Schlüssel haben, finden Sie unter the section called “Bestimmen des Zugriffs”.

Themen

- Anzeigen einer Schlüsselrichtlinie (Konsole)
- Anzeigen einer Schlüsselrichtlinie (AWS KMS-API)

Anzeigen einer Schlüsselrichtlinie (Konsole)

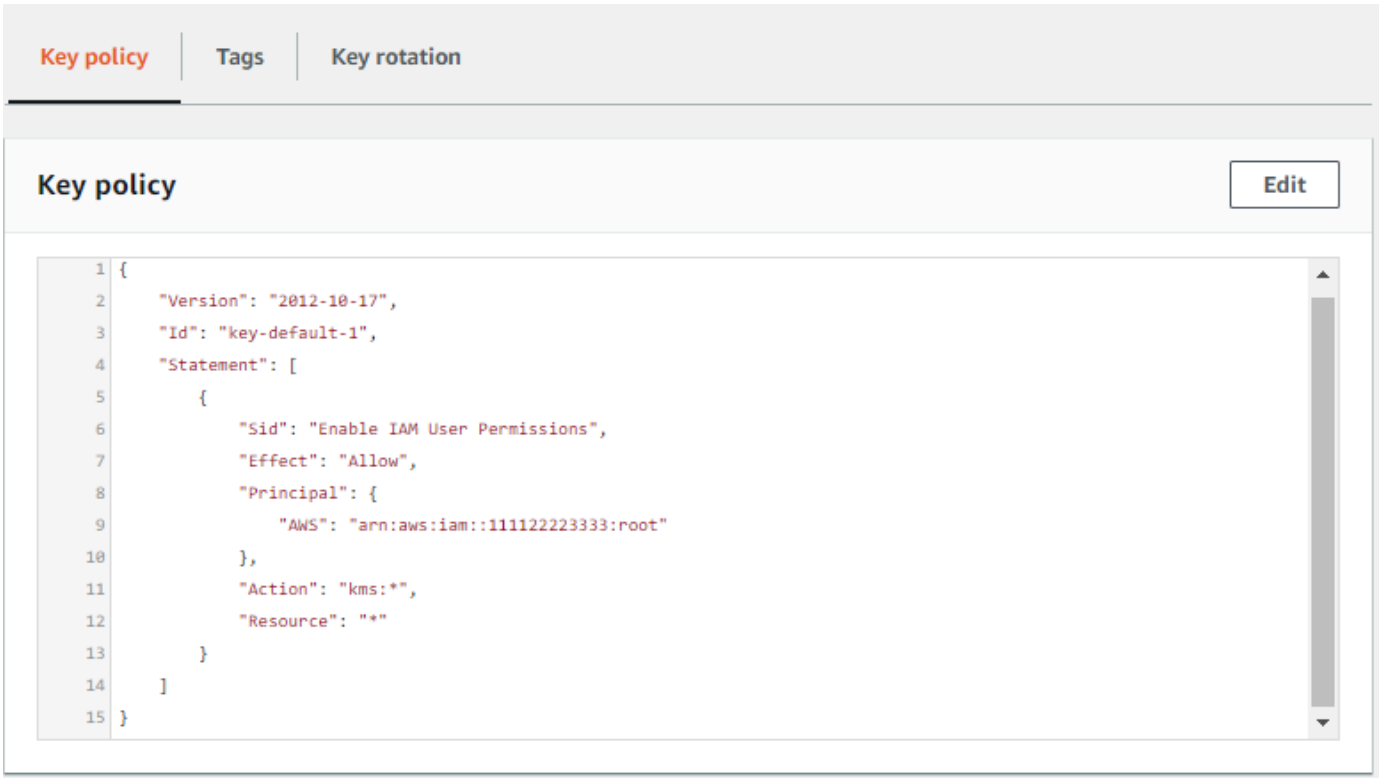
Autorisierte Benutzer können die Schlüsselrichtlinie für einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel auf der Registerkarte Key policy (Schlüsselrichtlinie) der AWS Management Console anzeigen.

Um die Schlüsselrichtlinie für einen KMS-Schlüssel in der anzuzeigenAWS Management Console, benötigen Sie die Berechtigungen kms:ListAliases, kms:DescribeKey und kms:GetKeyPolicy .

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus.
4. Wählen Sie in der Liste der KMS-Schlüssel den Alias oder die Schlüssel-ID des KMS-Schlüssels aus, den Sie untersuchen möchten.
5. Wählen Sie die Registerkarte Key policy (Schlüsselrichtlinie).

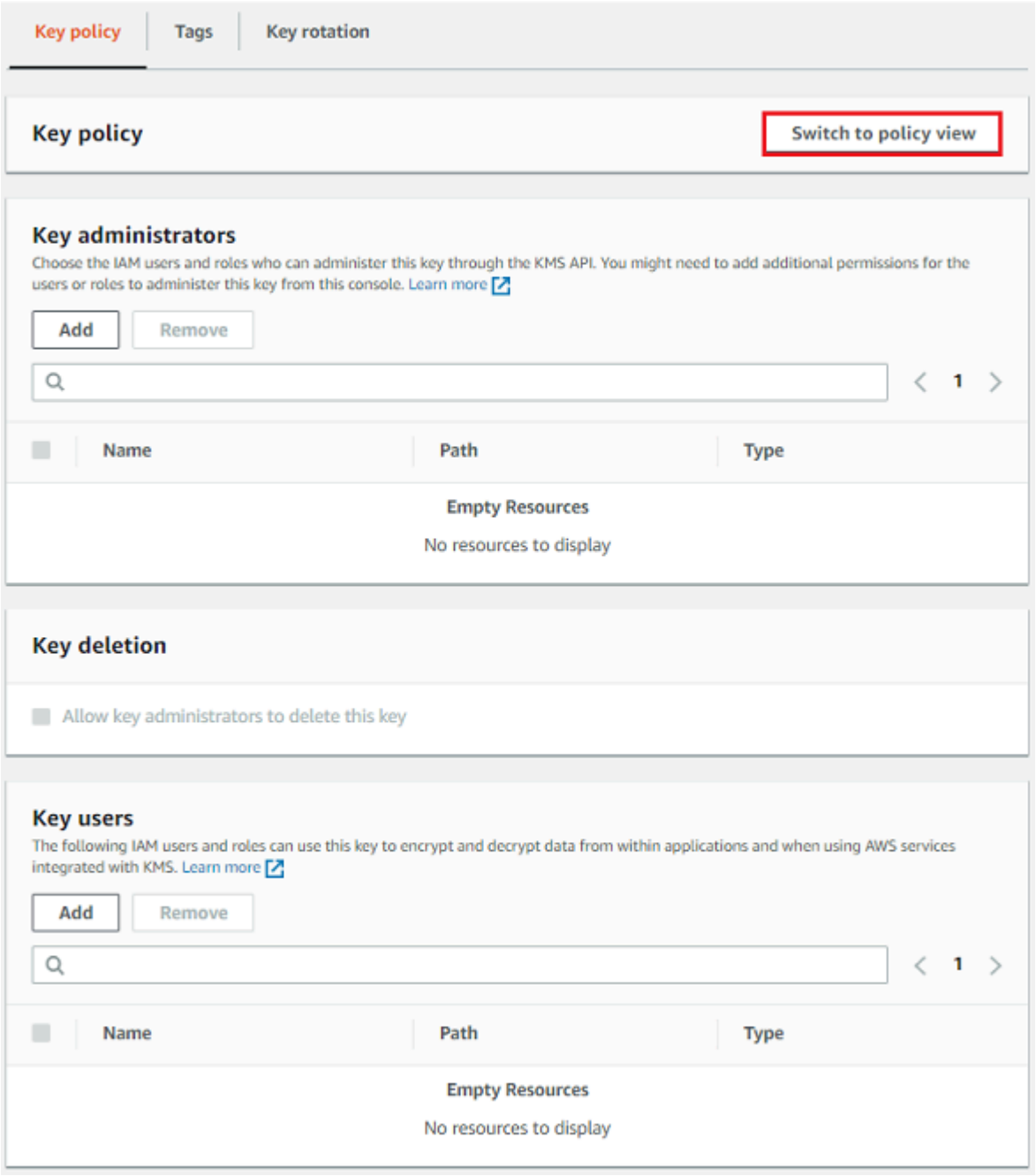
   Auf der Registerkarte Key policy (Schlüsselrichtlinie) wird möglicherweise das Schlüsselrichtliniendokument angezeigt. Dies ist eine Richtlinienansicht. In den Schlüsselrichtlinienanweisungen sehen Sie die Prinzipale, denen die Schlüsselrichtlinie Zugriff auf den KMS-Schlüssel gewährt hat, und Sie können die Aktionen anzeigen, die sie ausführen können.

   Das folgende Beispiel zeigt die Richtlinienansicht für die Standardschlüsselrichtlinie.



Andernfalls wird, wenn Sie den KMS-Schlüssel in AWS Management Console erstellt haben, die Standardansicht mit Bereichen für Key administrators (Schlüsseladministratoren) Key deletion (Schlüssellöschung) und Key Users (Schlüsselbenutzer) angezeigt. Wählen Sie zum Anzeigen der Schlüsselrichtliniendokuments Switch to policy view (Zur Richtlinienansicht wechseln).

Das folgende Beispiel zeigt die Standardansicht für die Standardschlüsselrichtlinie.



Anzeigen einer Schlüsselrichtlinie (AWS KMS-API)

Um die Schlüsselrichtlinie für einen KMS-Schlüssel in Ihrem abzurufenAWS-Konto, verwenden Sie die -GetKeyPolicyOperation in der AWS KMS-API. Sie können diesen Vorgang nicht verwenden, um eine Schlüsselrichtlinie in einem anderen Konto anzuzeigen.

Im folgenden Beispiel wird der get-key-policy Befehl in der AWS Command Line Interface (AWS CLI) verwendet, aber Sie können ein beliebiges AWS SDK verwenden, um diese Anforderung zu stellen.

Beachten Sie, dass der Parameter `PolicyName` erforderlich ist, obwohl `default` der einzige gültige Wert ist. Außerdem fordert dieser Befehl die Ausgabe als Text anstelle im JSON-Format an, damit sie leichter zu lesen ist.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige aus Ihrem Konto.

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

Die Antwort sollte der folgenden ähneln, die die Standardschlüsselrichtlinie zurückgibt.

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

Ändern einer Schlüsselrichtlinie

Sie können die Schlüsselrichtlinie für einen KMS-Schlüssel in Ihrem ändern, AWS-Konto indem Sie die PutKeyPolicy Operation AWS Management Console oder verwenden. Sie können diese Methoden nicht verwenden, um die Schlüsselrichtlinie eines KMS-Schlüssels in einem anderen AWS-Konto zu ändern.

Beachten Sie beim Ändern einer Schlüsselrichtlinie die folgenden Regeln:

- Sie können die Schlüsselrichtlinie für einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel anzeigen, aber ändern können Sie nur die Schlüsselrichtlinie für einen vom Kunden verwalteten Schlüssel. Die Richtlinien von Von AWS verwaltete Schlüssel werden von dem AWS-Service erstellt und verwaltet, der den KMS-Schlüssel in Ihrem Konto erstellt hat. Sie können die Schlüsselrichtlinie für einen AWS-eigener Schlüssel nicht anzeigen oder ändern.

- Sie können IAM-Benutzer, IAM-Rollen und AWS-Konten (Root-Benutzer) zur Schlüsselrichtlinie hinzufügen oder daraus entfernen und die Aktionen ändern, die diesen Prinzipalen erlaubt oder verweigert werden. Weitere Informationen zu den Möglichkeiten, um Prinzipale und Berechtigungen in einer Schlüsselrichtlinie festzulegen, finden Sie unter Schlüsselrichtlinien.
- Sie können keine IAM-Gruppen zu einer Schlüsselrichtlinie hinzufügen, aber Sie können mehrere IAM-Benutzer und IAM-Rollen hinzufügen. Weitere Informationen finden Sie unter Mehreren IAM-Prinzipalen Zugriff auf einen KMS-Schlüssel gewähren.
- Wenn Sie externe AWS-Konten zu einer Schlüsselrichtlinie hinzufügen, müssen Sie auch IAM-Richtlinien in den externen Konten verwenden, um IAM-Benutzern, -Gruppen oder -Rollen in diesen Konten Berechtigungen zu gewähren. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.
- Das sich ergebende Schlüssel-Richtliniendokument darf 32 KB (32.768 Bytes) nicht überschreiten.

Themen

So Ändern Sie eine Schlüsselrichtlinie:

Es gibt drei Möglichkeiten zum Ändern einer Schlüsselrichtlinie, die in den folgenden Abschnitten erläutert werden.

Themen

Verwendung der AWS Management Console-Standardansicht

Sie können die Konsole verwenden, um eine Schlüsselrichtlinie mit einer grafischen Benutzeroberfläche mit dem Namen Standardansicht zu ändern.

Wenn die folgenden Schritte nicht dem entsprechen, was Sie in der Konsole sehen, kann dies bedeuten, dass diese Schlüsselrichtlinie nicht durch die Konsole erstellt wurde. Es kann auch bedeuten, dass die Schlüsselrichtlinie auf eine Weise geändert wurde, die von der Standardansicht

der Konsole nicht unterstützt wird. Befolgen Sie in diesem Fall die Schritte unter Verwendung der AWS Management Console-Richtlinienansicht oder Verwenden der AWS KMS-API.

1. Zeigen Sie die Schlüsselrichtlinie für einen kundenverwalteten KMS-Schlüssel wie unter Anzeigen einer Schlüsselrichtlinie (Konsole) beschrieben an. (Sie können die Schlüsselrichtlinien von Von AWS verwaltete Schlüssel nicht ändern.)
2. Entscheiden Sie, was geändert werden soll.

   - Um Schlüsseladministratoren hinzuzufügen oder zu entfernen und festzulegen, ob sie den KMS-Schlüssel löschen dürfen oder nicht, verwenden Sie die Steuerelemente im Bereich Key Administrators (Schlüsseladministratoren) der Seite. Schlüsseladministratoren verwalten den KMS-Schlüssel. Dies umfasst die Aktivierung und Deaktivierung, das Festlegen der Schlüsselrichtlinie und das Aktivieren der Schlüsseldrehung.
   - Um Schlüsselbenutzer hinzuzufügen oder zu entfernen und festzulegen, ob externe AWS-Konten den KMS-Schlüssel verwenden dürfen oder nicht, verwenden Sie die Steuerelemente im Bereich Key users (Schlüsselbenutzer) auf der Seite. Schlüsselbenutzer können den KMS-Schlüssel in kryptografischen Produktionen verwenden, wie beispielsweise der Verschlüsselung, Entschlüsselung, erneuten Verschlüsselung und Generierung von Datenschlüsseln.

Verwendung der AWS Management Console-Richtlinienansicht

Sie können die Konsole verwenden, um ein Schlüsselrichtliniendokument über die Richtlinienansicht der Konsole zu ändern.

1. Zeigen Sie die Schlüsselrichtlinie für einen kundenverwalteten KMS-Schlüssel wie unter Anzeigen einer Schlüsselrichtlinie (Konsole) beschrieben an. (Sie können die Schlüsselrichtlinien von Von AWS verwaltete Schlüssel nicht ändern.)
2. Wählen Sie im Abschnitt Key Policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
3. Bearbeiten Sie das Schlüsselrichtliniendokument und wählen Sie dann Save Changes (Änderungen speichern) aus.

Verwenden der AWS KMS-API

Sie können die -PutKeyPolicyOperation verwenden, um die Schlüsselrichtlinie eines KMS-Schlüssels in Ihrem zu ändernAWS-Konto. Sie können diese API nicht für einen KMS-Schlüssel in einem anderen AWS-Konto verwenden.

1. Verwenden Sie die -GetKeyPolicyOperation, um das vorhandene Schlüsselrichtliniendokument abzurufen, und speichern Sie dann das Schlüsselrichtliniendokument in einer -Datei. Beispielcode in mehreren Programmiersprachen finden Sie unter Abrufen einer Schlüsselrichtlinie.
2. Öffnen Sie das Schlüsselrichtliniendokument in Ihrem bevorzugten Texteditor, bearbeiten Sie das Schlüsselrichtliniendokument und speichern Sie dann die Datei.
3. Verwenden Sie die -PutKeyPolicyOperation, um das aktualisierte Schlüsselrichtliniendokument auf den KMS-Schlüssel anzuwenden. Beispielcode in mehreren Programmiersprachen finden Sie unter Einstellen einer Schlüsselrichtlinie.

Ein Beispiel für das Kopieren einer Schlüsselrichtlinie von einem KMS-Schlüssel in einen anderen finden Sie im GetKeyPolicy Beispiel in der -AWS CLIBefehlsreferenz.

Mehreren IAM-Prinzipalen Zugriff auf einen KMS-Schlüssel gewähren

IAM-Gruppen sind keine gültigen Prinzipale in einer Schlüsselrichtlinie. Um mehreren Benutzern und Rollen den Zugriff auf einen KMS-Schlüssel zu erlauben, führen Sie einen der folgenden Schritte aus:

- Verwenden Sie eine IAM-Rolle als Prinzipal in der Schlüsselrichtlinie. Je nach Bedarf können mehrere autorisierte Benutzer die Rolle übernehmen. Weitere Informationen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.

  Sie können zwar mehrere IAM-Benutzer in einer Schlüsselrichtlinie auflisten, diese Vorgehensweise wird jedoch nicht empfohlen, da Sie die Schlüsselrichtlinie jedes Mal aktualisieren müssen, wenn sich die Liste der autorisierten Benutzer ändert. Bewährte IAM-Methoden raten außerdem von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.
- Verwenden Sie eine IAM-Richtlinie, um einer IAM-Gruppe eine Berechtigung zu erteilen. Stellen Sie dazu sicher, dass die Schlüsselrichtlinie die Anweisung enthält, die es IAM-Richtlinien ermöglicht, den Zugriff auf den KMS-Schlüssel zu erlauben, eine IAM-Richtlinie zu erstellen, die den Zugriff auf den KMS-Schlüssel erlaubt, diese Richtlinie dann einer IAM-Gruppe anzuhängen,

die die autorisierten IAM-Benutzer enthält. Bei diesem Ansatz müssen Sie keine Richtlinien ändern, wenn sich die Liste der autorisierten Benutzer ändert. Stattdessen müssen Sie einfach nur die Benutzer zu der entsprechenden IAM-Gruppe hinzufügen oder daraus entfernen. Weitere Informationen finden Sie unter IAM-Benutzergruppen im IAM-Benutzerhandbuch.

Weitere Informationen darüber, wie AWS KMS-Schlüsselrichtlinien und IAM-Richtlinien zusammen funktionieren, finden Sie unter Fehlerbehebung beim Schlüsselzugriff.

Berechtigungen für AWS Dienste in wichtigen Richtlinien

Viele AWS Dienste verwenden sie AWS KMS keys , um die von ihnen verwalteten Ressourcen zu schützen. Wenn ein Service AWS-eigene Schlüssel oder Von AWS verwaltete Schlüssel verwendet, legt der Service die Schlüsselrichtlinien für diese KMS-Schlüssel fest und verwaltet sie.

Wenn Sie jedoch einen vom Kunden verwalteten Schlüssel mit einem AWS -Service verwenden, legen Sie die Schlüsselrichtlinie fest und verwalten sie. Diese Schlüsselrichtlinie muss dem Service die Mindestberechtigungen gewähren, die er benötigt, um die Ressource in Ihrem Namen zu schützen. Wir empfehlen, dem Prinzip der geringsten Berechtigung zu folgen: Gewähren Sie dem Service nur die Berechtigungen, die er benötigt. Dazu sollten Sie ermitteln, welche Berechtigungen der Service benötigt, und die globalen Bedingungsschlüssel von AWS sowie die Bedingungsschlüssel von AWS KMS verwenden, um die Berechtigungen detaillierter festzulegen.

Um die Berechtigungen zu ermitteln, die der Service für einen vom Kunden verwalteten Schlüssel benötigt, lesen Sie in der Verschlüsselungsdokumentation für den Service nach. Die Berechtigungen, die Amazon Elastic Block Store (Amazon EBS) benötigt, finden Sie beispielsweise unter Berechtigungen für IAM-Benutzer im Amazon-EC2-Benutzerhandbuch für Linux-Instances und im Amazon-EC2-Benutzerhandbuch für Windows-Instances. Die Berechtigungen, die Secrets Manager benötigt, finden Sie unter Autorisieren der Nutzung des KMS-Schlüssels im AWS Secrets Manager - Benutzerhandbuch.

Implementieren der geringsten Berechtigungen

Wenn Sie einem AWS Dienst die Erlaubnis zur Verwendung eines KMS-Schlüssels erteilen, stellen Sie sicher, dass die Berechtigung nur für die Ressourcen gilt, auf die der Dienst in Ihrem Namen zugreifen muss. Diese Strategie der geringsten Rechte trägt dazu bei, die unbefugte Verwendung eines KMS-Schlüssels zu verhindern, wenn Anfragen zwischen AWS Diensten weitergeleitet werden.

Um eine Strategie mit den geringsten Rechten zu implementieren, empfehlen wir die Verwendung von Bedingungsschlüsseln für den AWS KMS Verschlüsselungskontext und den globalen Quell-ARN- oder Quellkonto-Bedingungsschlüssel.

Verwenden von Verschlüsselungskontext-Bedingungsschlüsseln

Die effektivste Methode zur Implementierung von Berechtigungen mit den geringsten Rechten bei der Nutzung von AWS KMS Ressourcen besteht darin, die kms:EncryptionContextKeysBedingungsschlüssel kms:EncryptionContext:context-keyoder in die Richtlinie aufzunehmen, die es den Prinzipalen ermöglicht, AWS KMS kryptografische Operationen aufzurufen. Diese Bedingungsschlüssel sind besonders effektiv, da sie die Berechtigung mit dem Verschlüsselungskontext verknüpfen, der an den Chiffretext gebunden ist, wenn die Ressource verschlüsselt wird.

Verwenden Sie Schlüssel für Bedingungen für den Verschlüsselungskontext nur, wenn es sich bei der Aktion in der Richtlinienanweisung um eine AWS KMS symmetrische kryptografische Operation handelt, die einen `EncryptionContext` Parameter benötigt, z. B. Operationen wie CreateGrantoder Decrypt. GenerateDataKey (Eine Liste der unterstützten Operationen finden Sie unter kms:EncryptionContext:context-key oder kms:EncryptionContextKeys.) Wenn Sie diese Bedingungsschlüssel verwenden, um z. B. andere Operationen zuzulassen, wird der DescribeKeyZugriff verweigert.

Legen Sie den Wert auf den Verschlüsselungskontext fest, den der Service beim Verschlüsseln der Ressource verwendet. Diese Informationen finden Sie normalerweise im Kapitel zur Sicherheit in der Service-Dokumentation. Beispielsweise identifiziert der Verschlüsselungskontext für AWS Proton die AWS Proton-Ressource und die zugehörige Vorlage. Der Verschlüsselungskontext für AWS Secrets Manager identifiziert das Geheimnis und seine Version. Der Verschlüsselungskontext für Amazon Location identifiziert den Tracker oder die Erfassung.

Die folgende Schlüsselrichtlinien-Beispielanweisung ermöglicht es Amazon Location Service, Erteilungen im Namen autorisierter Benutzer zu erstellen. Diese Richtlinienanweisung schränkt die Berechtigung ein, indem sie die Schlüssel kms: ViaService, kms: und `kms:EncryptionContext:`*`context-key`* condition verwendetCallerAccount, um die Berechtigung an eine bestimmte Tracker-Ressource zu binden.

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

Verwenden der Bedingungsschlüssel **`aws:SourceArn`** oder **`aws:SourceAccount`**

Wenn der Prinzipal in einer Schlüsselrichtlinien-Anweisung ein AWS -Service-Prinzipal ist, empfehlen wir dringend, zusätzlich zum Bedingungsschlüssel `kms:EncryptionContext:`*`context-key`* die globalen Bedingungsschlüssel aws:SourceArn oder aws:SourceAccount zu verwenden. Die ARN- und Kontowerte sind nur dann im Autorisierungskontext enthalten, wenn eine Anfrage AWS KMS von einem anderen AWS Dienst eingeht. Diese Kombination von Bedingungen implementiert die geringsten Berechtigungen und verhindert ein potenzielles Szenario des verwirrten Stellvertreters. Dienstprinzipale werden normalerweise nicht als Prinzipale in einer wichtigen Richtlinie verwendet, aber für einige AWS Dienste, z. B. AWS CloudTrail, ist dies erforderlich.

Um die globalen Bedingungsschlüssel `aws:SourceArn` oder `aws:SourceAccount` zu verwenden, legen Sie den Wert auf den Amazon-Ressourcennamen (ARN) oder das Konto der Ressource fest, die verschlüsselt wird. In einer Schlüsselrichtlinien-Anweisung, die AWS CloudTrail die Berechtigung zum Verschlüsseln eines Trail gibt, legen Sie den Wert von `aws:SourceArn` auf den ARN des Trail fest. Nutzen Sie, wann immer möglich, den spezifischeren Wert `aws:SourceArn`. Legen Sie den Wert auf den ARN oder ein ARN-Muster mit Platzhalterzeichen fest. Wenn Sie den ARN der Ressource nicht kennen, verwenden Sie stattdessen `aws:SourceAccount`.

> **Note**
>
> Wenn ein Ressourcen-ARN Zeichen enthält, die in einer AWS KMS Schlüsselrichtlinie nicht zulässig sind, können Sie diesen Ressourcen-ARN nicht im Wert des `aws:SourceArn` Bedingungsschlüssels verwenden. Verwenden Sie stattdessen den Bedingungsschlüssel `aws:SourceAccount`. Weitere Informationen zu Dokumentenregeln für Schlüsselrichtlinien finden Sie unter Schlüsselrichtlinienformat.

In der folgenden Beispiel-Schlüsselrichtlinie ist der Prinzipal, der die Berechtigungen erhält, der AWS CloudTrail -Service-Prinzipal `cloudtrail.amazonaws.com`. Um die geringsten Berechtigungen zu implementieren, verwendet diese Richtlinie die Bedingungsschlüssel `aws:SourceArn` und `kms:EncryptionContext:`*`context-key`* Die Richtlinienanweisung ermöglicht CloudTrail die Verwendung des KMS-Schlüssels zur Generierung des Datenschlüssels, der zur Verschlüsselung eines Trails verwendet wird. Die Bedingungen `aws:SourceArn` und `kms:EncryptionContext:`*`context-key`* werden unabhängig ausgewertet. Jede Anforderung, den KMS-Schlüssel für die angegebene Produktion zu verwenden, muss beide Bedingungen erfüllen.

Um die Berechtigung des Services auf den `finance`-Trail im Beispielkonto (111122223333) und die Region `us-west-2` zu beschränken, legt diese Richtlinienanweisung den Bedingungsschlüssel `aws:SourceArn` auf den ARN eines bestimmten Trail fest. Die Bedingungsanweisung verwendet den ArnEqualsOperator, um sicherzustellen, dass jedes Element im ARN beim Abgleich unabhängig ausgewertet wird. Das Beispiel verwendet den Bedingungsschlüssel `kms:EncryptionContext:`*`context-key`* auch, um die Berechtigung auf Trail in einem bestimmten Konto und einer bestimmten Region zu beschränken.

Bevor Sie diese Schlüsselrichtlinie verwenden, ersetzen Sie die Beispiel-Konto-ID, die Region und den Trail-Namen durch gültige Werte aus Ihrem Konto.

```
{
  "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/*"
          ]
```

```
        }
      }
    }
  ]
}
```

Verwenden von IAM-Richtlinien mit AWS KMS

Sie können IAM-Richtlinien zusammen mit wichtigen Richtlinien, Zuschüssen und VPC-Endpunktrichtlinien verwenden, um den Zugriff auf Ihr AWS KMS keys IN zu kontrollieren. AWS KMS

> **Note**
>
> Um mit einer IAM-Richtlinie den Zugriff auf einen KMS-Schlüssel zu steuern, muss die Schlüsselrichtlinie für den KMS-Schlüssel dem Konto die Berechtigung erteilen, IAM-Richtlinien zu nutzen. Insbesondere muss die Schlüsselrichtlinie die Richtlinienanweisung enthalten, die IAM-Richtlinien aktiviert.
> In diesem Abschnitt wird erklärt, wie Sie mithilfe von IAM-Richtlinien den Zugriff auf Vorgänge steuern können. AWS KMS Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.

Alle KMS-Schlüssel müssen eine Schlüsselrichtlinie haben. IAM-Richtlinien sind optional. Um mit einer IAM-Richtlinie den Zugriff auf einen KMS-Schlüssel zu steuern, muss die Schlüsselrichtlinie für den KMS-Schlüssel dem Konto die Berechtigung erteilen, IAM-Richtlinien zu nutzen. Insbesondere muss die Schlüsselrichtlinie die Richtlinienanweisung enthalten, die IAM-Richtlinien aktiviert.

IAM-Richtlinien können den Zugriff auf jeden AWS KMS Vorgang steuern. Im Gegensatz zu wichtigen Richtlinien können IAM-Richtlinien den Zugriff auf mehrere KMS-Schlüssel steuern und Berechtigungen für den Betrieb mehrerer verwandter AWS Dienste bereitstellen. IAM-Richtlinien sind jedoch besonders nützlich, um den Zugriff auf Vorgänge zu kontrollieren, z. B. solche CreateKey, die nicht durch eine Schlüsselrichtlinie gesteuert werden können, da sie keinen bestimmten KMS-Schlüssel beinhalten.

Wenn Sie AWS KMS über einen Amazon Virtual Private Cloud (Amazon VPC) -Endpunkt zugreifen, können Sie auch eine VPC-Endpunktrichtlinie verwenden, um den Zugriff auf Ihre AWS KMS Ressourcen zu beschränken, wenn Sie den Endpunkt verwenden. Wenn Sie beispielsweise den VPC-Endpunkt verwenden, gestatten Sie möglicherweise nur den Prinzipalen in Ihrem AWS-Konto

Zugriff auf Ihre vom Kunden verwalteten Schlüssel. Details hierzu finden Sie unter Steuern des Zugriffs auf einen VPC-Endpunkt.

Hilfe beim Schreiben und Formatieren eines JSON-Richtliniendokuments finden Sie in der IAM-JSON-Richtlinienreferenz im IAM-Benutzerhandbuch.

Themen

Übersicht über IAM-Richtlinien

Sie können IAM-Richtlinien auf folgende Weisen verwenden:

- Anfügen einer Berechtigungsrichtlinie an eine Rolle für Verbund oder kontoübergreifende Berechtigungen – Sie können einer IAM-Rolle eine IAM-Richtlinie anfügen, um einen Identitätsverbund zu aktivieren, kontoübergreifende Berechtigungen zu erlauben oder Anwendungen, die auf EC2-Instances ausgeführt werden, Berechtigungen zu gewähren. Weitere Informationen zu den verschiedenen Anwendungsfällen für IAM-Rollen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.
- Anfügen einer Berechtigungsrichtlinie an einen Benutzer oder ein Gruppe – Sie können eine Richtlinie anfügen, die es einem Benutzer oder einer Gruppe von Benutzern erlaubt, AWS KMS -Operationen aufzurufen. In den bewährten Methoden von IAM wird jedoch empfohlen, nach Möglichkeit Identitäten mit temporären Anmeldeinformationen, z. B. IAM-Rollen, zu verwenden.

Das folgende Beispiel zeigt eine IAM-Richtlinie mit Berechtigungen. AWS KMS Diese Richtlinie erlaubt es den IAM-Identitäten, denen sie angefügt ist, eine Liste aller KMS-Schlüssel und -Aliasen abzurufen.

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

```
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

Wie alle IAM-Richtlinien enthält diese Richtlinie kein `Principal`-Element. Wenn Sie einer IAM-Identität eine IAM-Richtlinie anfügen, erhält diese Identität die in der Richtlinie angegebenen Berechtigungen.

Eine Tabelle mit allen AWS KMS API-Aktionen und den Ressourcen, für die sie gelten, finden Sie unterBerechtigungsreferenz.

Bewährte Methoden für IAM-Richtlinien

Die Sicherung des Zugriffs auf AWS KMS keys ist entscheidend für die Sicherheit all Ihrer AWS Ressourcen. KMS-Schlüssel werden verwendet, um viele der sensibelsten Ressourcen in Ihrem zu schützen AWS-Konto. Nehmen Sie sich Zeit, um die Schlüsselrichtlinien, IAM-Richtlinien, Erteilungen und VPC-Endpunktrichtlinien zu entwerfen, die den Zugriff auf Ihre KMS-Schlüssel steuern.

Wenden Sie in IAM-Richtlinienanweisungen, die den Zugriff auf KMS-Schlüssel steuern, das Prinzip der geringsten Berechtigung an. Geben Sie IAM-Prinzipalen nur die Berechtigungen ein, die sie nur für die KMS-Schlüssel benötigen, die sie verwenden oder verwalten müssen.

Die folgenden bewährten Methoden gelten für IAM-Richtlinien, die den Zugriff auf AWS KMS Schlüssel und Aliase steuern. Eine allgemeine Anleitung zu bewährten IAM-Richtlinien finden Sie unter Bewährte Methoden zur Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden von Schlüsselrichtlinien

> Stellen Sie nach Möglichkeit Berechtigungen, die sich auf einen KMS-Schlüssel auswirken, in einer Schlüsselrichtlinie bereit anstatt in einer IAM-Richtlinie, die auf viele KMS-Schlüssel angewendet werden kann, einschließlich derjenigen in anderen AWS-Konten. Dies ist besonders wichtig für vertrauliche Berechtigungen wie kms: PutKeyPolicy und kms:, ScheduleKeyDeletion aber auch für kryptografische Operationen, die bestimmen, wie Ihre Daten geschützt werden.

Beschränken Sie die Erlaubnis CreateKey

Erteilen Sie nur den Prinzipalen die Erlaubnis, Schlüssel (kms: CreateKey) zu erstellen, die sie benötigen. Prinzipale, die einen KMS-Schlüssel erstellen, legen auch seine Schlüsselrichtlinie fest, damit sie sich selbst und anderen die Berechtigung zur Verwendung und Verwaltung der von ihnen erstellten KMS-Schlüssel erteilen können. Wenn Sie diese Berechtigung erlauben, sollten Sie sie vielleicht mithilfe von -Richtlinienbedingungen beschränken. Sie können beispielsweise die KeySpec Bedingung kms: verwenden, um die Erlaubnis auf KMS-Schlüssel mit symmetrischer Verschlüsselung zu beschränken.

Angeben von KMS-Schlüsseln in einer IAM-Richtlinie

Geben Sie als bewährte Methode den Schüssel-ARN jedes KMS-Schlüssels an, für den die Berechtigung im `Resource`-Element der Richtlinienanweisung gilt. Diese Methode beschränkt die Berechtigung auf die KMS-Schlüssel, die vom Prinzipal benötigt werden. Dieses `Resource`-Element listet beispielsweise nur den KMS-Schlüssel auf, den der Prinzipal verwenden muss.

```
"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"
]
```

Wenn die Angabe von KMS-Schlüsseln nicht praktikabel ist, verwenden Sie einen `Resource` Wert, der den Zugriff auf KMS-Schlüssel in einer vertrauenswürdigen AWS-Konto Region einschränkt, z. B. `arn:aws:kms:`*`region`*`:`*`account`*`:key/*` Oder beschränken Sie den Zugriff auf KMS-Schlüssel in allen Regionen (*) einer vertrauenswürdigen Person AWS-Konto, z. B. `arn:aws:kms:*:`*`account`*`:key/*`

Sie können keine Schlüssel-ID, Aliasnamen oder Alias-ARN verwenden, um einen KMS-Schlüssel in dem `Resource`-Feld einer IAM-Richtlinie darzustellen. Wenn Sie einen Alias-ARN angeben, gilt die Richtlinie für den Alias und nicht für den KMS-Schlüssel. Weitere allgemeine Informationen zu IAM-Richtlinien für Aliase finden Sie unter Steuern des Zugriffs auf Aliasse

Vermeiden Sie "Resource": "*" in einer IAM-Richtlinie

Verwenden Sie Platzhalterzeichen (*) umsichtig. In einer Schlüsselrichtlinie stellt das Platzhalterzeichen in dem `Resource`-Element den KMS-Schlüssel dar, an den die Schlüsselrichtlinie angefügt ist. In einer IAM-Richtlinie wendet jedoch nur ein Platzhalterzeichen im `Resource` Element (`"Resource": "*"`) die Berechtigungen auf alle KMS-Schlüssel an, zu deren Verwendung AWS-Konten das Konto des Prinzipalbenutzers berechtigt ist. Dies kann

KMS-Schlüssel in anderen Bereichen AWS-Konten sowie KMS-Schlüssel im Konto des Prinzipals umfassen.

Um beispielsweise einen KMS-Schlüssel in einem anderen zu verwenden AWS-Konto, benötigt ein Principal die Genehmigung durch die Schlüsselrichtlinie des KMS-Schlüssels im externen Konto und durch eine IAM-Richtlinie in seinem eigenen Konto. Angenommen, ein beliebiges Konto erteilte Ihrem AWS-Konto die kms:Decrypt-Berechtigung für seine KMS-Schlüssel. In diesem Fall würde eine IAM-Richtlinie in Ihrem Konto, die einer Rolle die `kms:Decrypt`-Berechtigung für alle KMS-Schlüssel (`"Resource": "*"`) erteilt, den IAM-Teil der Anforderung erfüllen. Daher können Prinzipale, die diese Rolle übernehmen können, nun Chiffretexte mit dem KMS-Schlüssel im nicht-vertrauenswürdigen Konto entschlüsseln. Einträge für ihre Operationen werden in den CloudTrail Protokollen beider Konten angezeigt.

Vermeiden Sie insbesondere die Verwendung von `"Resource": "*"` in einer Richtlinienanweisung, die die folgenden API-Operationen zulässt. Diese Operationen können für KMS-Schlüssel in anderen Ländern aufgerufen werden AWS-Konten.

- DescribeKey
- GetKeyRotationStatus
- Kryptografische Operationen (Verschlüsseln, Entschlüsseln,,, GenerateDataKey,, GenerateDataKeyPair, GenerateDataKeyWithoutPlaintextGenerateDataKeyPairWithoutPlaintext, Signieren GetPublicKeyReEncrypt, Überprüfen)
- CreateGrant, ListGrants, ListRetirableGrants, RetireGrant, RevokeGrant

Wann "Resource": "*" verwendet werden sollte

Verwenden Sie in einer IAM-Richtlinie ein Platzhalterzeichen im `Resource`-Element nur für Berechtigungen, die es erfordern. Nur die folgenden Berechtigungen erfordern das `"Resource": "*"`-Element.

- km: CreateKey
- km: GenerateRandom
- km: ListAliases
- km: ListKeys
- Berechtigungen für benutzerdefinierte Schlüsselspeicher wie kms: CreateCustomKeyStore und kms: ConnectCustomKeyStore.

> **Note**
>
> Berechtigungen für Aliasoperationen (kms: CreateAlias, kms: UpdateAlias, kms: DeleteAlias) müssen an den Alias und den KMS-Schlüssel angehängt werden. Sie können die `"Resource": "*"` in einer IAM-Richtlinie verwenden, um die Aliase und die KMS-Schlüssel darzustellen, oder geben Sie die Aliase und KMS-Schlüssel im `Resource`-Element an. Beispiele finden Sie unter Steuern des Zugriffs auf Aliasse.

Die Beispiele in diesem Thema enthalten weitere Informationen und Anleitungen zum Entwerfen von IAM-Richtlinien für KMS-Schlüssel. Allgemeine Hinweise zu AWS KMS bewährten Verfahren finden Sie in den AWS Key Management Service Best Practices (PDF). Bewährte IAM-Methoden für alle AWS Ressourcen finden Sie unter Bewährte Methoden zur Sicherheit in IAM im IAM-Benutzerhandbuch.

Angeben von KMS-Schlüsseln in IAM-Richtlinienanweisungen

Sie können eine IAM-Richtlinie verwenden, um einem Prinzipal die Verwendung oder Verwaltung von KMS-Schlüsseln zu erlauben. KMS-Schlüssel werden in dem `Resource`-Element der Richtlinienanweisung angegeben.

- Um einen KMS-Schlüssel in einer IAM-Richtlinienanweisung anzugeben, müssen Sie dessen Schüssel-ARN verwenden. Sie können keine Schlüssel-ID und keinen Aliasnamen oder Alias-ARN verwenden, um einen KMS-Schlüssel in einer IAM-Richtlinienanweisung zu identifizieren.

  Zum Beispiel: „`Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`“

  Um den Zugriff auf einen KMS-Schlüssel anhand seiner Aliase zu steuern, verwenden Sie die Bedingungsschlüssel kms: RequestAlias oder kms:. ResourceAliases Details hierzu finden Sie unter ABAC für AWS KMS.

  Verwenden Sie einen Alias-ARN nur als Ressource in einer Richtlinienanweisung, die den Zugriff auf Aliasoperationen wie CreateAliasUpdateAlias, oder steuert DeleteAlias. Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.
- Um mehrere KMS-Schlüssel im Konto und in der Region anzugeben, verwenden Sie Platzhalterzeichen (*) in den Regions- oder Ressourcen-ID-Positionen des Schlüssel-ARN.

Um beispielsweise alle KMS-Schlüssel in der Region USA West (Oregon) eines Kontos anzugeben, verwenden Sie `"Resource": "arn:aws:kms:us-west-2:111122223333:key/*"`. Um alle KMS-Schlüssel in allen Regionen des Kontos anzugeben, verwenden Sie `"Resource": "arn:aws:kms:*:111122223333:key/*"`.

- Um alle KMS-Schlüssel darzustellen, verwenden Sie nur ein Platzhalterzeichen (`"*"`). Verwenden Sie dieses Format für Operationen, die keinen bestimmten KMS-Schlüssel verwenden, nämlich CreateKeyGenerateRandom, ListAliases, und ListKeys.

Wenn Sie Ihre Richtlinienanweisungen schreiben, ist es eine bewährte Methode, nur die KMS-Schlüssel anzugeben, die der Prinzipal verwenden muss, anstatt ihm Zugriff auf alle KMS-Schlüssel zu gewähren.

Die folgende IAM-Richtlinienanweisung ermöglicht es dem Principal beispielsweise DescribeKey, die GenerateDataKey, Decrypt-Operationen nur für die KMS-Schlüssel aufzurufen, die im `Resource` Element der Richtlinienanweisung aufgeführt sind. Durch die bewährte Methode, KMS-Schlüssel nach dem Schlüssel-ARN anzugeben, wird sichergestellt, dass die Berechtigungen nur auf die angegebenen KMS-Schlüssel beschränkt sind.

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

Um die Berechtigung auf alle KMS-Schlüssel in einem bestimmten vertrauenswürdigen Bereich anzuwenden AWS-Konto, können Sie Platzhalterzeichen (*) an den Positionen Region und Schlüssel-ID verwenden. Beispielsweise erlaubt die folgende Richtlinienanweisung es dem Prinzipal,

die angegebenen Operationen für alle KMS-Schlüssel in zwei vertrauenswürdigen Beispielkonten aufzurufen.

```
{
  "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/*"
    ]
  }
}
```

Sie können auch nur ein Platzhalterzeichen (`"*"`) in dem `Resource`-Element verwenden. Da es den Zugriff auf alle KMS-Schlüssel erlaubt, die das Konto nutzen darf, wird es in erster Linie für Operationen ohne einen bestimmten KMS-Schlüssel und für `Deny`-Anweisungen empfohlen. Sie können es auch in Richtlinienanweisungen verwenden, die nur weniger vertrauliche schreibgeschützte Operationen erlauben. Um festzustellen, ob ein AWS KMS Vorgang einen bestimmten KMS-Schlüssel beinhaltet, suchen Sie in der Spalte Ressourcen der Tabelle unter nach dem KMS-Schlüsselwert. the section called “Berechtigungsreferenz”

Die folgende Richtlinienanweisung nutzt einen `Deny`-Effekt, um Prinzipalen die Nutzung der angegebenen Operationen auf irgendeinem KMS-Schlüssel zu verweigern. Sie verwendet ein Platzhalterzeichen in dem `Resource`-Element, um alle KMS-Schlüssel darzustellen.

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

```
    "Resource": "*"
  }
}
```

Die folgende Richtlinienanweisung verwendet nur ein Platzhalterzeichen, um alle KMS-Schlüssel darzustellen. Sie erlaubt jedoch nur weniger sensible schreibgeschützte Operationen und Operationen, die nicht für einen bestimmten KMS-Schlüssel gelten.

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

Für die Verwendung der AWS KMS Konsole sind Berechtigungen erforderlich

Um mit der AWS KMS Konsole arbeiten zu können, müssen Benutzer über Mindestberechtigungen verfügen, die es ihnen ermöglichen, mit den AWS KMS Ressourcen in ihrer Konsole zu arbeiten AWS-Konto. Zusätzlich zu diesen AWS KMS -Berechtigungen müssen Benutzer auch über die Berechtigungen zum Auflisten von IAM-Benutzern und IAM-Rollen verfügen. Wenn Sie eine IAM-Richtlinie erstellen, die restriktiver ist als die erforderlichen Mindestberechtigungen, funktioniert die AWS KMS Konsole für Benutzer mit dieser IAM-Richtlinie nicht wie vorgesehen.

Die Mindestberechtigungen, um einem Benutzer reinen Lesezugriff auf die AWS KMS -Konsole zu gewähren, finden Sie unter Erlauben Sie einem Benutzer, KMS-Schlüssel in der AWS KMS Konsole anzuzeigen.

Damit Benutzer bei der Erstellung und Verwaltung von KMS-Schlüsseln mit der AWS KMS Konsole arbeiten können, fügen Sie dem Benutzer die AWSKeyManagementServicePowerUserverwaltete Richtlinie an, wie im folgenden Abschnitt beschrieben.

Für Benutzer, die mit der AWS KMS -API über die AWS -SDKs, AWS Command Line Interface oder AWS Tools for PowerShell. Sie müssen diesen Benutzern jedoch die Berechtigung zur Verwendung der API erteilen. Weitere Informationen finden Sie unter Berechtigungsreferenz.

AWS verwaltete Richtlinie für Hauptbenutzer

Sie können eine von `AWSKeyManagementServicePowerUser` verwaltete Richtlinie verwenden, um IAM-Prinzipalen in Ihrem Konto die Berechtigungen eines Hauptbenutzers zu gewähren. Hauptbenutzer können KMS-Schlüssel erstellen, die von ihnen erstellten KMS-Schlüssel verwenden und verwalten sowie alle KMS-Schlüssel und IAM-Identitäten anzeigen. Benutzer mit der von `AWSKeyManagementServicePowerUser` verwalteten Richtlinie können auch Berechtigungen aus anderen Quellen erhalten, darunter Schlüsselrichtlinien, andere IAM-Richtlinien und Erteilungen.

`AWSKeyManagementServicePowerUser`ist eine AWS verwaltete IAM-Richtlinie. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie unter AWS Verwaltete Richtlinien im IAM-Benutzerhandbuch.

> **Note**
>
> In dieser Richtlinie enthaltene Berechtigungen, die für einen KMS-Schlüssel spezifisch sind, wie z. B. `kms:TagResource` und`kms:GetKeyRotationStatus`, sind nur wirksam, wenn die Schlüsselrichtlinie für diesen KMS-Schlüssel ausdrücklich die Verwendung von IAM-Richtlinien AWS-Konto zur Steuerung des Zugriffs auf den Schlüssel zulässt. Um festzustellen, ob sich eine Berechtigung auf einen KMS-Schlüssel bezieht, siehe AWS KMS Berechtigungen und suchen Sie in der Spalte Ressourcen nach dem Wert KMS-Schlüssel. Mit dieser Richtlinie erhält ein Hauptbenutzer Berechtigungen für jeden KMS-Schlüssel mit einer Schlüsselrichtlinie, die den Vorgang zulässt. Für kontoübergreifende Berechtigungen wie `kms:DescribeKey` und `kms:ListGrants` kann dies KMS-Schlüssel in nicht vertrauenswürdigen AWS-Konten einschließen. Details dazu finden Sie unter Bewährte Methoden für IAM-Richtlinien und Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben. Um festzustellen, ob eine Berechtigung für KMS-Schlüssel in anderen Konten gültig ist, siehe AWS KMS Berechtigungen und suchen Sie in der Spalte Kontoübergreifende Verwendung nach dem Wert Ja.
> Damit die Hauptbenutzer die AWS KMS Konsole ohne Fehler aufrufen können, benötigt der Principal das Tag: GetResources permission, das nicht in der `AWSKeyManagementServicePowerUser` Richtlinie enthalten ist. Sie können diese Berechtigung in einer separaten IAM-Richtlinie erlauben.

Die AWSKeyManagementServicePowerUserverwaltete IAM-Richtlinie umfasst die folgenden Berechtigungen.

- Erlaubt es Prinzipalen, KMS-Schlüssel zu erstellen. Da dieser Prozess das Festlegen der Schlüsselrichtlinie beinhaltet, können Hauptbenutzer sich selbst und anderen die Berechtigung zur Verwendung und Verwaltung der von ihnen erstellten KMS-Schlüssel erteilen.
- Erlaubt es Benutzern, Aliase und Tags auf allen KMS-Schlüsseln zu erstellen und zu löschen. Wenn Sie ein Tag oder einen Alias ändern, wird die Berechtigung zur Verwendung und Verwaltung des KMS-Schlüssels erteilt oder verweigert. Details hierzu finden Sie unter ABAC für AWS KMS.
- Erlaubt es Benutzern, detaillierte Informationen zu allen KMS-Schlüsseln zu erhalten, einschließlich ihres Schlüssel-ARN, der kryptografischen Konfiguration, der Schlüsselrichtlinie, Aliase, Tags und Drehungsstatus.
- Erlaubt es Benutzern, IAM-Benutzer, -Gruppen und -Rollen aufzulisten.
- Diese Richtlinie erlaubt es Prinzipalen nicht, KMS-Schlüssel zu verwenden oder zu verwalten, die sie nicht erstellt haben. Sie können jedoch Aliase und Tags auf allen KMS-Schlüsseln ändern, was ihnen die Berechtigung zur Verwendung oder Verwaltung eines KMS-Schlüssels erlauben oder verweigern kann.

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

```
    ]
}
```

Beispiele für IAM-Richtlinien

In diesem Abschnitt finden Sie Beispiele für IAM-Richtlinien, die Berechtigungen für diverse AWS KMS -Aktionen gewähren.

> **Important**
>
> Einige der Berechtigungen in den folgenden Richtlinien sind nur erlaubt, wenn sie auch in der Schlüsselrichtlinie des KMS-Schlüssels erlaubt werden. Weitere Informationen finden Sie unter Berechtigungsreferenz.

Hilfe beim Schreiben und Formatieren eines JSON-Richtliniendokuments finden Sie in der IAM-JSON-Richtlinienreferenz im IAM-Benutzerhandbuch.

Beispiele

- Erlauben Sie einem Benutzer, KMS-Schlüssel in der AWS KMS Konsole anzuzeigen
- Einem Benutzer das Erstellen von KMS-Schlüsseln erlauben
- Ermöglicht einem Benutzer das Verschlüsseln und Entschlüsseln mit einem beliebigen KMS-Schlüssel in einem bestimmten AWS-Konto
- Erlaubt einem Benutzer das Verschlüsseln und Entschlüsseln mit einem beliebigen KMS-Schlüssel in einer bestimmten Region AWS-Konto
- Einem Benutzer das Verschlüsseln und Entschlüsseln mit bestimmten KMS-Schlüssel erlauben
- Einen Benutzer am Deaktivieren oder Löschen von KMS-Schlüsseln hindern

Erlauben Sie einem Benutzer, KMS-Schlüssel in der AWS KMS Konsole anzuzeigen

Die folgende IAM-Richtlinie ermöglicht Benutzern nur Lesezugriff auf die Konsole. AWS KMS Benutzer mit diesen Berechtigungen können alle KMS-Schlüssel in ihrer Datenbank einsehen AWS-Konto, sie können jedoch keine KMS-Schlüssel erstellen oder ändern.

Um KMS-Schlüssel auf den Seiten Von AWS verwaltete Schlüsselund Vom Kunden verwaltete Schlüssel anzeigen zu können, benötigen Prinzipale die GetResources Berechtigungen kms:

ListAliases, kms: und tag:, auch wenn die Schlüssel keine Tags oder Aliase haben. ListKeys Die verbleibenden Berechtigungen, insbesondere kms: DescribeKey, sind erforderlich, um optionale Spalten und Daten der KMS-Schlüsseltabelle auf den KMS-Schlüsseldetailseiten anzuzeigen. Die ListRoles Berechtigungen iam: ListUsers und iam: sind erforderlich, um die Schlüsselrichtlinie in der Standardansicht fehlerfrei anzuzeigen. Um Daten auf der Seite Benutzerdefinierte Schlüsselspeicher und Details zu KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern anzuzeigen, benötigen Prinzipale außerdem die kms: -Berechtigung. DescribeCustomKeyStores

Wenn Sie den Konsolenzugriff eines Benutzers auf bestimmte KMS-Schlüssel beschränken, zeigt die Konsole einen Fehler für jeden KMS-Schlüssel an, der nicht sichtbar ist.

Diese Richtlinie umfasst zwei Richtlinienanweisungen. Das `Resource`-Element in der ersten Richtlinienanweisung gewährt die angegebenen Berechtigungen für alle KMS-Schlüssel in allen Regionen des Beispiel- AWS-Konto. Konsolenbetrachter benötigen keinen zusätzlichen Zugriff, da die AWS KMS -Konsole nur KMS-Schlüssel im Konto des Prinzipals anzeigt. Dies gilt auch dann, wenn sie berechtigt sind, KMS-Schlüssel in anderen AWS-Konten Datenbanken einzusehen. Für die verbleibenden Berechtigungen AWS KMS und die IAM-Berechtigungen ist ein `"Resource": "*"` Element erforderlich, da sie für keinen bestimmten KMS-Schlüssel gelten.

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

Einem Benutzer das Erstellen von KMS-Schlüsseln erlauben

Die folgende IAM-Richtlinie erlaubt Benutzern die Erstellung aller Arten von KMS-Schlüsseln. Der Wert des `Resource` Elements ist darauf * zurückzuführen, dass der `CreateKey` Vorgang keine bestimmten AWS KMS Ressourcen (KMS-Schlüssel oder Aliase) verwendet.

Um den Benutzer auf bestimmte Typen von KMS-Schlüsseln zu beschränken, verwenden Sie die KeyOrigin Bedingungsschlüssel kms: KeyUsage, kms: und kms:. KeySpec

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

Prinzipale, die Schlüssel erstellen, benötigen möglicherweise einige verwandte Berechtigungen.

- kms: PutKeyPolicy — Prinzipale, die über die entsprechenden `kms:CreateKey` Rechte verfügen, können die Richtlinie für den ersten Schlüssel für den KMS-Schlüssel festlegen. Der `CreateKey` Aufrufer muss jedoch über die PutKeyPolicykms-Berechtigung verfügen, mit der er die KMS-Schlüsselrichtlinie ändern kann, oder er muss den `BypassPolicyLockoutSafetyCheck` Parameter von angeben`CreateKey`, was nicht empfohlen wird. Der `CreateKey`-Anrufer kann die `kms:PutKeyPolicy`-Berechtigung für den KMS-Schlüssel aus einer IAM-Richtlinie erhalten, oder er kann diese Berechtigung in die Schlüsselrichtlinie des KMS-Schlüssels aufnehmen, den er erstellt.
- kms: TagResource — Um dem KMS-Schlüssel während des `CreateKey` Vorgangs Tags hinzuzufügen, muss der `CreateKey` Aufrufer in einer IAM-Richtlinie über die kms: TagResource -Berechtigung verfügen. Die Aufnahme dieser Berechtigung in die Schlüsselrichtlinie des neuen

KMS-Schlüssels ist nicht ausreichend. Wenn der `CreateKey`-Anrufer jedoch `kms:TagResource` in die ursprünglichen Schlüsselrichtlinie aufnimmt, kann er Tags in einem separaten Aufruf hinzufügen, nachdem der KMS-Schlüssel erstellt wurde.

- kms: CreateAlias — Principals, die einen KMS-Schlüssel in der AWS KMS Konsole erstellen, müssen über die CreateAlias kms-Berechtigung für den KMS-Schlüssel und den Alias verfügen. (Die Konsole führt zwei Aufrufe aus; einen an `CreateKey` und einen an `CreateAlias`). Sie müssen die Aliasberechtigung in einer IAM-Richtlinie angeben. Sie können dem KMS-Schlüssel die Berechtigungen in einer Schlüsselrichtlinie oder einer IAM-Richtlinie bereitstellen. Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.

Darüber hinaus gewährt die folgende IAM-Richtlinie `kms:TagResource` Berechtigungen für alle KMS-Schlüssel im Konto AWS-Konto und `kms:CreateAlias` Berechtigungen für alle Aliase des Kontos. `kms:CreateKey` Sie enthält auch einige nützliche schreibgeschützte Berechtigungen, die nur in einer IAM-Richtlinie bereitgestellt werden können.

Diese IAM-Richtlinie enthält nicht `kms:PutKeyPolicy` oder andere Berechtigungen, die in einer Schlüsselrichtlinie festgelegt werden können. Dabei handelt es sich um eine bewährte Methode, diese Berechtigungen in der Schlüsselrichtlinie festzulegen, in der sie ausschließlich für einen KMS-Schlüssel gelten.

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

Ermöglicht einem Benutzer das Verschlüsseln und Entschlüsseln mit einem beliebigen KMS-Schlüssel in einem bestimmten AWS-Konto

Die folgende IAM-Richtlinie ermöglicht es einem Benutzer, Daten mit einem beliebigen KMS-Schlüssel in 111122223333 zu verschlüsseln und zu entschlüsseln. AWS-Konto

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

Erlaubt einem Benutzer das Verschlüsseln und Entschlüsseln mit einem beliebigen KMS-Schlüssel in einer bestimmten Region AWS-Konto

Die folgende IAM-Richtlinie ermöglicht es einem Benutzer, Daten mit einem beliebigen KMS-Schlüssel AWS-Konto 111122223333 in der Region USA West (Oregon) zu verschlüsseln und zu entschlüsseln.

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

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

Einem Benutzer das Verschlüsseln und Entschlüsseln mit bestimmten KMS-Schlüssel erlauben

Die folgende IAM-Richtlinie erlaubt es einem Benutzer, mit den beiden im `Resource`-Element angegebenen KMS-Schlüssel Daten zu verschlüsseln und zu entschlüsseln. Um einen KMS-Schlüssel in einer IAM-Richtlinienanweisung anzugeben, müssen Sie dessen Schüssel-ARN verwenden.

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

Einen Benutzer am Deaktivieren oder Löschen von KMS-Schlüsseln hindern

Die folgende IAM-Richtlinie verhindert die Deaktivierung oder Löschung von KMS-Schlüsseln durch Benutzer, auch wenn eine andere IAM-Richtlinie oder eine Schlüsselrichtlinie diese Berechtigungen gewährt. Eine Richtlinie, die Berechtigungen explizit verweigert, hat Vorrang vor allen anderen Richtlinien, auch wenn diese dieselben Berechtigungen explizit erteilen. Weitere Informationen finden Sie unter Fehlerbehebung beim Schlüsselzugriff.

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

```
    "Effect": "Deny",
    "Action": [
      "kms:DisableKey",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

Erteilungen in AWS KMS

Eine Erteilung ist ein Richtlinieninstrument, das es AWS-Prinzipalen erlaubt, KMS-Schlüssel in kryptografischen Operationen zu verwenden. Es kann ihnen auch erlauben, einen KMS-Schlüssel anzuzeigen (`DescribeKey`) und Erteilungen zu erstellen und zu verwalten. Bei der Autorisierung des Zugriffs auf einen KMS-Schlüssel werden Erteilungen zusammen mit Schlüsselrichtlinien und IAM-Richtlinien berücksichtigt. Erteilungen werden häufig für temporäre Berechtigungen verwendet, da Sie eine erstellen, deren Berechtigungen verwenden und sie dann wieder löschen können, ohne Ihre Schlüsselrichtlinien oder IAM-Richtlinien zu ändern.

Erteilungen werden häufig von AWS-Services verwendet, die sich in AWS KMS integrieren, um Ihre ruhenden Daten zu verschlüsseln. Der Service erstellt eine Erteilung im Namen eines Benutzers im Konto, verwendet seine Berechtigungen und hebt die Erteilung auf, sobald die Aufgabe abgeschlossen ist. Weitere Informationen darüber, wie AWS-Services Erteilungen verwenden, finden Sie unter Verwendung von AWS KMS durch AWS-Service oder dem Thema Verschlüsselung im Ruhezustand im Benutzerhandbuch oder Entwicklerhandbuch für den Service.

Code-Beispiele zur Veranschaulichung der Arbeit mit Erteilungen in mehreren Programmiersprachen finden Sie unter Arbeiten mit Erteilungen.

Themen

- Informationen über Erteilungen
- Konzepte für Erteilungen
- Bewährte Methoden für AWS KMS-Erteilungen
- Erstellen einer Erteilung
- Verwalten von Erteilungen

Informationen über Erteilungen

Erteilungen sind ein sehr flexibler und nützlicher Zugriffsteuerungs-Mechanismus. Wenn Sie eine Erteilung für einen KMS-Schlüssel erstellen, erlaubt die Erteilung es dem Empfänger-Prinzipal, die angegebenen Erteilungs-Operationen für den KMS-Schlüssel aufzurufen, vorausgesetzt, dass alle in der Erteilung angegebenen Bedingungen erfüllt sind.

- Jede Erteilung erlaubt den Zugriff auf genau einen KMS-Schlüssel. Sie können eine Erteilung für einen KMS-Schlüssel in einem anderen AWS-Konto erstellen.
- Eine Erteilung kann den Zugriff auf einen KMS-Schlüssel erlauben, aber nicht den Zugriff verweigern.
- Jede Erteilung hat einen Empfänger-Prinzipal. Der Empfänger-Prinzipal kann im selben AWS-Konto wie der KMS-Schlüssel oder in einem anderen Konto eine oder mehrere Identitäten repräsentieren.
- Eine Erteilung kann nur Erteilungs-Operationen erlauben. Die Erteilungs-Operationen müssen durch den KMS-Schlüssel in der Erteilung unterstützt werden. Wenn Sie eine nicht unterstützte Operation angeben, schlägt die CreateGrant Anforderung mit einer `ValidationError` Ausnahme fehl.
- Der Empfänger-Prinzipal können die Berechtigungen verwenden, die ihnen durch die Erteilung gewährt werden, ohne die Erteilung anzugeben, genauso wie wenn die Berechtigungen aus einer Schlüsselrichtlinie oder IAM-Richtlinie stammen. Da die AWS KMS-API jedoch einem Konsistenzmodell folgt, kann es bei der Erstellung, Außerbetriebnahme oder Widerruf einer Erlaubnis zu einer kurzen Verzögerung kommen, bevor die Änderung in ganz AWS KMS verfügbar ist. Um die Berechtigungen in einer Erteilung sofort zu verwenden, verwenden Sie einen Erteilungs-Token.
- Ein autorisierter Prinzipal kann die Erteilung löschen, (aufheben oder widerrufen). Durch das Löschen einer Erteilung entfallen alle Berechtigungen, die durch die Erteilung erlaubt wurden. Sie müssen nicht herausfinden, welche Richtlinien hinzugefügt oder entfernt werden sollen, um die Erteilung rückgängig zu machen.
- AWS KMS begrenzt die Anzahl der Erteilungen für jeden KMS-Schlüssel. Details hierzu finden Sie unter Erteilungen pro KMS-Schlüssel: 50 000.

Seien Sie vorsichtig, wenn Sie Erteilungen erstellen und anderen die Berechtigung zum Erstellen von Erteilungen erteilen. Die Berechtigung zum Erstellen von Erteilungen hat Auswirkungen auf die

Sicherheit, ähnlich wie das Zulassen von kms:PutKeyPolicy die Berechtigung zum Festlegen von Richtlinien.

- Benutzer mit der Berechtigung zum Erstellen von Erteilungen für einen KMS-Schlüssel (`kms:CreateGrant`) können eine Erteilung verwenden, um es Benutzern und Rollen, einschließlich AWS-Services, zu erlauben, den KMS-Schlüssel zu verwenden. –Die Prinzipale können Identitäten in Ihrem eigenen AWS-Konto oder Identitäten in einem anderen Konto oder einer anderen Organisation sein.
- Erteilungen können nur eine Teilmenge von AWS KMS-Operationen verwenden. Sie können Erteilungen verwenden, um es Prinzipalen zu erlauben, den KMS-Schlüssel anzuzeigen, ihn in kryptografischen Operationen zu verwenden und Erteilungen zu erstellen und außer Betrieb zu nehmen. Details dazu finden Sie unter Erteilungs-Operationen. Sie können auch Erteilungs-Einschränkungen verwenden, um die Berechtigungen in einer Erteilung für eine Schlüssel mit symmetrischer Verschlüsselung einzuschränken.
- Prinzipale können Berechtigung zum Erstellen von Erteilungen aus einer Schlüsselrichtlinie oder IAM-Richtlinie erhalten. Prinzipale, die `kms:CreateGrant`-Berechtigung aus einer Richtlinie bekommen, können Zuschüsse für alle Erteilungs-Operationen auf dem KMS-Schlüssel erstellen. Diese Prinzipale müssen nicht über die Berechtigung verfügen, die sie für den Schlüssel erteilen. Wenn Sie die `kms:CreateGrant`-Berechtigung in einer Richtlinie erlauben, können Sie Richtlinienbedingungen verwenden, um diese Berechtigung einzuschränken.
- Prinzipale können auch die Berechtigung erhalten, Erteilungen aus einer Erteilung zu erstellen. Diese Prinzipale können nur die Berechtigungen delegieren, die ihnen erteilt wurden, selbst wenn sie über andere Berechtigungen aus einer Richtlinie verfügen. Details hierzu finden Sie unter Gewähren von CreateGrant Berechtigungen.

Hilfe zu Konzepten im Zusammenhang mit Erteilungen finden Sie unter Terminologie für Erteilungen.

Konzepte für Erteilungen

Um Erteilungen effektiv nutzen zu können, müssen Sie die Begriffe und Konzepte verstehen, die AWS KMS verwendet.

Einschränkungen für Erteilungen

> Eine Bedingung, die die Berechtigungen in der Erteilung einschränkt. Derzeit unterstützt AWS KMS Erteilungs-Einschränkungen anhand des Verschlüsselungskontexts in der Anforderung

für eine kryptografische Produktion. Details hierzu finden Sie unter Verwenden von Erteilungs-Einschränkungen.

Erteilungs-ID

Die eindeutige ID einer Erteilung für einen KMS-Schlüssel. Sie können eine Erteilungs-ID zusammen mit einer Schlüsselkennung verwenden, um eine Erteilung in einer - oder -RevokeGrantAnforderung zu identifizieren. RetireGrant

Erteilungs-Operationen

Die AWS KMS-Operationen, die Sie in einer Erteilung erlauben können. Wenn Sie andere Operationen angeben, schlägt die CreateGrant Anforderung mit einer `ValidationError` Ausnahme fehl. Dies sind auch die Operationen, die einen Erteilungs-Token akzeptieren. Ausführliche Informationen über diese Berechtigungen finden Sie unter AWS KMS Berechtigungen.

Diese Erteilungs-Operationen stellen tatsächlich die Berechtigung zur Verwendung der Produktion dar. Daher gilt, dass Sie für die `ReEncrypt`-Produktion `ReEncryptFrom`, `ReEncryptTo`, oder beide `ReEncrypt*` angeben können.

Die Erteilungs-Operationen sind:

- Kryptografische Operationen
  - Decrypt
  - Encrypt
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyPairWithoutPlaintext
  - GenerateDataKeyWithoutPlaintext
  - GenerateMac
  - ReEncryptFrom
  - ReEncryptTo
  - Sign
  - Verify
  - VerifyMac
- Andere Produktionen
  - CreateGrant

- DescribeKey
- GetPublicKey
- RetireGrant

Die Erteilungs-Operationen, die Sie erlauben, müssen durch den KMS-Schlüssel in der Erteilung unterstützt werden. Wenn Sie eine nicht unterstützte Operation angeben, schlägt die CreateGrant Anforderung mit einer `ValidationError` Ausnahme fehl. Beispielsweise können Erteilungen für KMS-Schlüssel mit symmetrischer Verschlüsselung die Operationen Signieren, Überprüfen, `GenerateMac` oder `VerifyMac` nicht zulassen. Erteilungen für asymmetrische KMS-Schlüssel dürfen keine Operationen zulassen, die Datenschlüssel oder Datenschlüsselpaare generieren.

Erteilungs-Token

Die AWS KMS-API folgt einem eventuellen Konsistenzmodell. Wenn Sie eine Erteilung erstellen, kann es zu einer kurzen Verzögerung kommen, bis die Änderung in AWS KMS verfügbar ist. In der Regel dauert es weniger als ein paar Sekunden, bis sich die Änderung im gesamten System verbreitet, in einigen Fällen kann es jedoch mehrere Minuten dauern. Wenn Sie versuchen, eine Erteilung zu verwenden, bevor sie sich vollständig im System verbreitet hat, erhalten Sie möglicherweise eine Zugriffsverweigerungsmeldung. Mit einem Erteilungs-Token können Sie auf die Erteilung verweisen und die Erteilungs-Berechtigungen sofort verwenden.

Ein Berechtigungserteilungs-Token ist eine eindeutige, nicht-geheime, base64-kodierte Zeichenfolge mit variabler Länge, die eine Erteilung darstellt. Sie können den Erteilungs-Token verwenden, um die Erteilung in jeder Erteilungs-Produktion zu identifizieren. Da der Token-Wert jedoch ein Hash-Digest ist, zeigt er keine Details über die Erteilung an.

Ein Erteilungs-Token darf nur so lange verwendet werden, bis sich die Erteilung vollständig in AWS KMS ausgebreitet hat. Danach kann der Empfänger-Prinzipal die Berechtigung in der Erteilung ohne Angabe eines Grant-Token oder eines anderen Beweises für die Erteilung verwenden. Sie können jederzeit ein Erteilungs-Token verwenden, aber sobald die Erteilung die letztendliche Konsistenz erreicht hat, verwendet AWS KMS die Erteilung, um Berechtigungen zu bestimmen, und nicht das Erteilungs-Token.

Mit dem folgenden Befehl wird beispielsweise die -GenerateDataKeyOperation aufgerufen. Er verwendet ein Erteilung-Token, um die Erteilung darzustellen, die dem Aufrufer (dem erteilenden Prinzipal) die Berechtigung zum Aufrufen von `GenerateDataKey` für den angegebenen KMS-Schlüssel erteilt.

```
$ aws kms generate-data-key \
```

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

Sie können den Erteilungs-Token verwenden, um eine Erteilung in Operationen zu identifizieren, die Erteilungen verwalten. Beispielsweise kann der ausscheidende Prinzipal ein Erteilungs-Token in einem Aufruf der -RetireGrantOperation verwenden.

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

`CreateGrant` ist die einzige Produktion, die ein Erteilungs-Token zurückgibt. Sie können kein Erteilungs-Token von einer anderen AWS KMS Operation oder vom CloudTrail Protokollereignis für die CreateGrant Operation erhalten. Die ListRetirableGrants Operationen ListGrants und geben die Erteilungs-ID zurück, aber kein Erteilungs-Token.

Details hierzu finden Sie unter Verwenden eines Erteilungs-Token.

Erteilender Prinzipal

Die Identitäten, die die in der Erteilung angegebenen Berechtigungen erhalten. Jede Erteilung hat einen Empfänger-Prinzipal, aber der Empfänger-Prinzipal kann mehrere Identitäten repräsentieren.

Der Empfänger-Prinzipal kann ein beliebiger AWS-Prinzipal sein, wie ein AWS-Konto (Root), ein IAM-Benutzer, eine IAM-Rolle, eine Verbundrolle oder ein Verbundbenutzer oder ein Benutzer einer übernommenen Rolle. Der Empfänger-Prinzipal kann sich im selben Konto wie der KMS-Schlüssel oder in einem anderen Konto befinden. Der Empfänger-Prinzipal kann jedoch kein Service-Prinzipal und keine IAM-Gruppe oder AWS-Organisation sein.

> **Note**
>
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Außerbetriebnahme (einer Erteilung)

Beendet eine Erteilung. Sie können eine Erteilung aufheben, wenn Sie die Berechtigungen nicht mehr brauchen.

Wenn Sie eine Erteilung widerrufen oder außer Betrieb nehmen, wird die Erteilung gelöscht. Die Außerbetriebnahme erfolgt jedoch durch einen Prinzipal, der in der Erteilung angegeben ist. Das Widerrufen erfolgt in der Regel durch einen Schlüsseladministrator. Details hierzu finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.

Außerbetriebnahme eines Prinzipals

Ein Prinzipal, der eine Erteilung aufheben kann. Sie können einen ausscheidenden Prinzipal in einer Erteilung angeben, jedoch ist es nicht erforderlich. Der ausscheidende Prinzipal kann ein beliebiger AWS-Prinzipal sein, einschließlich AWS-Konten, IAM-Benutzer, IAM-Rollen, Verbundbenutzer und angenommener Rollenbenutzer. Der ausscheidende Prinzipal kann sich im selben Konto wie der KMS-Schlüssel oder in einem anderen Konto befinden.

> **Note**
>
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Zusätzlich zu dem in der Erteilung angegebenen ausscheidenden Prinzipal kann eine Erteilung von dem AWS-Konto aufgehoben werden, in dem die Erteilung erstellt wurde. Wenn die Erteilung die `RetireGrant`-Produktion erlaubt, kann der Empfänger-Prinzipal die Erteilung aufheben. Auch das AWS-Konto oder ein AWS-Konto, bei dem es sich um den ausscheidenden Prinzipal handelt, kann die Berechtigung zur Außerbetriebnahme einer Erteilung an einen IAM-Prinzipal in demselben AWS-Konto delegieren. Details hierzu finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.

Widerrufen (einer Erteilung)

Beendet eine Erteilung. Sie widerrufen eine Erteilung, um aktiv die Berechtigung abzulehnen, die die Erteilung erlaubt.

Wenn Sie eine Erteilung widerrufen oder außer Betrieb nehmen, wird die Erteilung gelöscht. Die Außerbetriebnahme erfolgt jedoch durch einen Prinzipal, der in der Erteilung angegeben ist. Das

Widerrufen erfolgt in der Regel durch einen Schlüsseladministrator. Details hierzu finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.

Letztendliche Konsistenz (für Erteilungen)

Die AWS KMS-API folgt einem eventuellen Konsistenzmodell. Wenn Sie eine Erteilung erstellen, aufheben oder widerrufen, kann es zu einer kurzen Verzögerung kommen, bevor die Änderung in allen Bereichen von AWS KMS verfügbar ist. In der Regel dauert es weniger als ein paar Sekunden, bis sich die Änderung im gesamten System verbreitet, in einigen Fällen kann es jedoch mehrere Minuten dauern.

Diese kurze Verzögerung kann Ihnen auffallen, wenn Sie unerwartete Fehler erhalten. Wenn Sie beispielsweise versuchen, eine neue Erteilung zu verwalten oder die Berechtigungen in einer neuen Erteilung zu verwenden, bevor die Erteilung in allen Bereichen von AWS KMS bekannt ist, wird möglicherweise eine Zugriff-verweigert-Fehlermeldung angezeigt. Wenn Sie eine Erteilung aufheben oder widerrufen, kann der Empfänger-Prinzipal seine Berechtigungen möglicherweise für einen kurzen Zeitraum verwenden, bis die Erteilung vollständig gelöscht wurde. Die typische Strategie besteht darin, die Anforderung erneut zu versuchen, und einige AWS-SDKs enthalten automatische Backoff- und Wiederholungslogik.

AWS KMS verfügt über Funktionen, um diese kurze Verzögerung zu verringern.

- Um die Berechtigungen in einer neuen Erteilung sofort zu verwenden, verwenden Sie einen Erteilungs-Token. Sie können einen Erteilungs-Token verwenden, um die Erteilung in jeder Erteilungs-Produktion zu identifizieren. Anweisungen finden Sie unter Verwenden eines Erteilungs-Token.
- Der CreateGrant Vorgang verfügt über einen `Name` Parameter, der verhindert, dass Wiederholungsvorgänge doppelte Erteilungen erstellen.

> **Note**
>
> Erteilungs-Token ersetzen die Gültigkeit der Erteilung, bis alle Endpunkte im Service mit dem neuen Erteilungsstatus aktualisiert wurden. In den meisten Fällen wird die letztendliche Konsistenz innerhalb von fünf Minuten erreicht.

Weitere Informationen finden Sie unter AWS KMS Letztendliche Konsistenz.

Bewährte Methoden für AWS KMS-Erteilungen

AWS KMS empfiehlt beim Erstellen, Verwenden und Verwalten von Erteilungen die folgenden bewährten Methoden.

- Beschränken Sie die Berechtigungen in der Erteilung auf diejenigen, die der Empfänger-Prinzipal benötigt. Erteilen Sie Zugriff nach dem Prinzip der geringsten Berechtigung.
- Verwenden Sie einen bestimmten Empfänger-Prinzipal, z. B. eine IAM-Rolle, und erteilen Sie dem Empfänger-Prinzipal nur die Berechtigung, die API-Operationen zu verwenden, die er benötigt.
- Verwenden Sie den Verschlüsselungskontext Erteilungs-Einschränkungen, um sicherzustellen, dass Anrufer den KMS-Schlüssel für den beabsichtigten Zweck verwenden. Einzelheiten zur Verwendung des Verschlüsselungskontexts in einer Anforderung zum Schutz Ihrer Daten finden Sie unter So schützen Sie die Integrität Ihrer verschlüsselten Daten mithilfe von AWS Key Management Service und EncryptionContext im AWS -Sicherheitsblog.

  > **Tip**
  >
  > Verwenden Sie nach Möglichkeit die EncryptionContextEqual Erteilungseinschränkung. Die EncryptionContextSubset Erteilungseinschränkung ist schwieriger zu verwenden. Wenn Sie sie verwenden müssen, lesen Sie die Dokumentation sorgfältig durch und testen Sie die Erteilungs-Einschränkung, um sicherzustellen, dass sie wie beabsichtigt funktioniert.

- Löschen Sie doppelte Erteilungen. Doppelte Erteilungen haben dieselben API-Aktionen und denselben Schlüssel-ARN, Empfänger-Prinzipal, Verschlüsselungskontext und Namen. Wenn Sie die ursprüngliche Erteilung aufheben oder widerrufen, aber die doppelte Erteilungen belassen, stellen die verbleibenden doppelten Erteilungen unbeabsichtigte Eskalationen von Rechten dar. Um bei `CreateGrant`-Anforderungen zu vermeiden, dass doppelte Erteilungen erstellt werden, verwenden Sie den `Name`-Parameter. Um doppelte Erteilungen zu erkennen, verwenden Sie die -ListGrantsOperation. Wenn Sie versehentlich eine doppelte Erteilung erstellen, widerrufen Sie bzw. nehmen Sie diese sie so schnell wie möglich außer Betrieb.

  > **Note**
  >
  > Erteilungen für AWS-verwaltete Schlüssel könnten wie Duplikate aussehen, haben aber unterschiedliche Empfänger-Prinzipale.
  > Das `GranteePrincipal`-Feld in der `ListGrants`-Antwort enthält normalerweise den Berechtigungsprinzipal der Genehmigung. Wenn der Empfänger-Prinzipal in der Erteilung

jedoch ein AWS-Service ist, enthält das `GranteePrincipal`-Feld den Service-Prinzipal, der mehrere verschiedene Empfänger-Prinzipale repräsentieren kann.

- Denken Sie daran, dass Erteilungen nicht automatisch ablaufen. Außerbetriebnahme oder Widerruf der Erteilung sobald die Berechtigung nicht mehr benötigt wird. Erteilungen, die nicht gelöscht werden, können ein Sicherheitsrisiko für verschlüsselte Ressourcen verursachen.

Erstellen einer Erteilung

Informieren Sie sich vor dem Erstellen einer Erteilung über die Optionen für benutzerdefinierte Erteilungen. Sie können auch Erteilungs-Einschränkungen verwenden, um die Berechtigungen in einer Erteilung einzuschränken. Erfahren Sie auch mehr über die Erteilung der `CreateGrant`-Berechtigung. Prinzipale, die die Berechtigung zum Erstellen von Erteilungen aus einer Erteilung erhalten, sind in den Erteilungen beschränkt, die sie erstellen können.

Themen

- Erstellen einer Erteilung
- Verwenden von Erteilungs-Einschränkungen
- Gewähren von CreateGrant Berechtigungen

Erstellen einer Erteilung

Um eine Erteilung zu erstellen, rufen Sie die CreateGrant Operation auf. Geben Sie einen KMS-Schlüssel an, einen Empfänger-Prinzipal und eine Liste der zulässigen Erteilungs-Operationen. Sie können auch einen optionalen ausscheidenden Prinzipal angeben. Um die Erteilung anzupassen, verwenden Sie optionale `Constraints`-Parameter, um Erteilungseinschränkungen zu definieren.

Wenn Sie eine Erteilung erstellen, aufheben oder widerrufen, kann es zu einer kurzen Verzögerung (in der Regel weniger als fünf Minuten) kommen, bevor die Änderung in allen Bereichen von AWS KMS verfügbar ist. Weitere Informationen finden Sie unter Letztendliche Konsistenz (für Erteilungen).

Der folgende `CreateGrant`-Befehl erstellt beispielsweise eine Erteilung, die Benutzer, die die `keyUserRole`-Rolle annehmen dürfen, den Aufruf der Decrypt-Operation für den angegebenen symmetrischen KMS-Schlüssel erlaubt. Die Erteilung legt mit dem Parameter `RetiringPrincipal` einen Prinzipal fest, der die erteilte Berechtigung aufheben kann. Es enthält auch eine Erteilungseinschränkung, die die Berechtigung nur zulässt, wenn der Verschlüsselungskontext in der Anforderung `"Department": "IT"` einschließt.

```
$  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}
```

Wenn Ihr Code die `CreateGrant`-Produktion erneut versucht, oder ein AWS-SDK verwendet, das Anforderungen automatisch wiederholt, verwenden Sie den optionalen Name-Parameter, um das Erstellen von doppelten Erteilungen zu verhindern. Wenn AWS KMS eine `CreateGrant`-Anforderung für eine Erteilung mit den gleichen Eigenschaften wie eine vorhandene Erteilung erhält, einschließlich des Namens, erkennt es die Anforderung als Wiederholungsversuch und erstellt keine neue Erteilung. Sie können nicht den `Name`-Wert verwenden, um die Erteilung in einer AWS KMS-Produktion zu identifizieren.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen im Namen der Erteilung an. Er kann in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

```
$ 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}
```

Code-Beispiele zur Veranschaulichung der Arbeit mit Erteilungen in mehreren Programmiersprachen finden Sie unter Arbeiten mit Erteilungen.

Verwenden von Erteilungs-Einschränkungen

Erteilungs-Einschränkungen legen Bedingungen für die Berechtigungen fest, die der Empfänger-Prinzipal ausführen kann. Erteilungs-Einschränkungen treten an die Stelle von Bedingungsschlüssel in einer Schlüsselrichtlinie oder IAM-Richtlinie. Jeder Erteilungs-Einschränkungs-Wert kann bis zu 8 Verschlüsselungskontext-Paare enthalten. Der Verschlüsselungskontext-Wert in jeder Erteilungs-Einschränkung darf 384 Zeichen nicht überschreiten.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

AWS KMS unterstützt zwei Erteilungs-Einschränkungen, `EncryptionContextEquals` und `EncryptionContextSubset`, die beide Anforderungen für den Verschlüsselungskontext in einer Anforderung für eine kryptografische Produktion festlegen.

Die Verschlüsselungskontext-Erteilungs-Einschränkungen sind für die Verwendung mit Erteilungs-Operationen entworfen, die einen Verschlüsselungskontext-Parameter haben.

- Beschränkungen für den Verschlüsselungskontext gelten nur in einer Erteilung für einen KMS-Schlüssel mit symmetrischer Verschlüsselung. Kryptografische Operationen mit asymmetrischen KMS-Schlüsseln unterstützen keinen Verschlüsselungskontext.
- Die Verschlüsselungskontext-Einschränkung wird für `DescribeKey`- und `RetireGrant`-Operationen ignoriert. `DescribeKey` und `RetireGrant` haben keinen Verschlüsselungskontext-Parameter, aber Sie können diese Operationen in eine Erteilung einschließen, die über eine Verschlüsselungskontext-Einschränkung verfügt.
- Sie können eine Verschlüsselungskontext-Einschränkung in einer Erteilung für die `CreateGrant`-Produktion verwenden. Die Verschlüsselungskontext-Einschränkung erfordert, dass alle Erteilungen, die mit der `CreateGrant`-Berechtigung erstellt wurden, eine ebenso strenge oder strengere Verschlüsselungskontext-Einschränkung haben.

AWS KMS unterstützt die folgenden Verschlüsselungskontext-Erteilungs-Einschränkungen.

EncryptionContextEquals

Verwenden Sie `EncryptionContextEquals`, um den exakten Verschlüsselungskontext für zulässige Anforderungen anzugeben.

`EncryptionContextEquals` erfordert, dass die Verschlüsselungskontext-Paare in der Anforderung genau mit den Verschlüsselungskontext-Paaren in der Erteilungs-Einschränkung übereinstimmen (inkl. Groß-/Kleinschreibung). Die Paare können in beliebiger Reihenfolge erscheinen. Die Schlüssel und Werte in den Paaren können nicht variieren.

Zum Beispiel, wenn die `EncryptionContextEquals`-Erteilungs-Einschränkung das `"Department": "IT"`-Verschlüsselungskontext-Paar erfordert, erlaubt die Erteilung Anforderungen des angegebenen Typs nur dann, wenn der Verschlüsselungskontext in der Anforderung genau `"Department": "IT"` ist.

EncryptionContextSubset

Verwenden Sie `EncryptionContextSubset`, um zu erfordern, dass Anforderungen bestimmte Verschlüsselungskontext-Paare enthalten.

`EncryptionContextSubset` erfordert, dass die Anforderung alle Verschlüsselungskontext-Paare in der Erteilungs-Einschränkung enthält (genaue Übereinstimmung, inkl. Groß-/Kleinschreibung), aber die Anforderung kann auch zusätzliche Verschlüsselungskontext-Paare enthalten. Die Paare können in beliebiger Reihenfolge erscheinen. Die Schlüssel und Werte in den Paaren können nicht variieren.

Zum Beispiel, wenn die `EncryptionContextSubset`-Erteilungs-Einschränkung das `Department=IT`-Verschlüsselungskontext-Paar erfordert, erlaubt die Erteilung Anforderungen des angegebenen Typs nur dann, wenn der Verschlüsselungskontext in der Anforderung `"Department": "IT"` ist oder `"Department": "IT"` enthält, zusammen mit anderen Verschlüsselungskontext-Paaren, z. B. `"Department": "IT","Purpose": "Test"`.

Um eine Einschränkung des Verschlüsselungskontexts in einer Erteilung für einen KMS-Schlüssel mit symmetrischer Verschlüsselung anzugeben, verwenden Sie den `-Constraints`Parameter in der -CreateGrantOperation. Die Erteilung, die dieser Befehl erstellt, gewährt Benutzern, die die `keyUserRole`-Rolle annehmen dürfen, die Erlaubnis, die Decrypt-Operation aufzurufen. Diese Berechtigung ist jedoch nur wirksam, wenn der Verschlüsselungskontext in der `Decrypt`-Anforderung ein `"Department": "IT"`-Verschlüsselungskontext-Paar ist.

```
$ 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}
```

Die resultierende Erteilung sieht wie die folgende aus. Beachten Sie, dass die erteilte Berechtigung für die `keyUserRole`-Rolle nur wirksam ist, wenn die `Decrypt`-Anforderung das

Verschlüsselungskontextpaar enthält, das in der Erteilungseinschränkung angegeben ist. Verwenden Sie die -ListGrantsOperation, um die Erteilungen für einen KMS-Schlüssel zu finden.

```
$ 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"
        }
    ]
}
```

Um die `EncryptionContextEquals`-Erteilungs-Einschränkung zu erfüllen, muss der Verschlüsselungskontext in der Anforderung für die `Decrypt`-Produktion ein `"Department": "IT"`-Paar sein. Eine Anforderung wie die folgende vom Empfänger-Prinzipal würde die `EncryptionContextEquals`-Erteilungs-Einschränkung erfüllen.

```
$ 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
```

Wenn die Erteilungs-Einschränkung `EncryptionContextSubset` ist, müssen die Verschlüsselungskontext-Paare in der Anforderung die Verschlüsselungskontext-Paare

in der Erteilungs-Einschränkung beinhalten, aber die Anforderung kann auch andere Verschlüsselungskontext-Paare enthalten. Die folgende Erteilungs-Einschränkung erfordert, dass eines der Verschlüsselungskontext-Paare in der Anforderung `"Deparment": "IT"` ist.

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

Eine Anforderung wie die folgende vom Empfänger-Prinzipal würde sowohl die `EncryptionContextEqual`- und `EncryptionContextSubset`-Erteilungs-Einschränkung in diesem Beispiel erfüllen.

```
$ 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
```

Eine Anforderung wie die folgende vom Empfänger-Prinzipal würde die `EncryptionContextSubset`-Erteilungs-Einschränkung erfüllen, aber nicht die `EncryptionContextEquals`-Erteilungs-Einschränkung.

```
$ 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 Services verwenden oft Verschlüsselungskontext-Einschränkungen in den Erteilungen, die ihnen die Berechtigung zur Verwendung von KMS-Schlüssel in Ihrem erteilen AWS-Konto. Beispielsweise verwendet Amazon DynamoDB eine Erteilung wie die folgende, um die Berechtigung zu erhalten, den Von AWS verwalteter Schlüssel für DynamoDB in Ihrem Konto zu verwenden. Die `EncryptionContextSubset`-Erteilungseinschränkung in dieser Erteilung macht die Berechtigungen in der Erteilung nur wirksam, wenn der Verschlüsselungskontext in der Anforderung `"subscriberID": "111122223333"`- und `"tableName": "Services"`-Paare enthält. Diese Erteilungs-Einschränkung bedeutet, dass die Erteilung es DynamoDB erlaubt, den angegebenen KMS-Schlüssel nur für eine bestimmte Tabelle in Ihrem AWS-Konto zu verwenden.

Um diese Ausgabe zu erhalten, führen Sie die -ListGrantsOperation auf der Von AWS verwalteter Schlüssel für DynamoDB in Ihrem Konto aus.

```
$ 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"
        }
    ]
}
```

Gewähren von CreateGrant Berechtigungen

Eine Erteilung kann die Berechtigung zum Aufrufen der `CreateGrant`-Produktion enthalten. Aber wenn ein Empfänger-Prinzipal die Berechtigung zum Aufrufen von `CreateGrant` von einer Erteilung erhält, und nicht von einer Richtlinie, ist diese Berechtigung eingeschränkt.

- Der Empfänger-Prinzipal kann nur Erteilungen erlauben, die einige oder alle Operationen in der übergeordneten Erteilung zulassen.
- Die Erteilungs-Einschränkungen in den Erteilungen, die sie erstellen, müssen mindestens so streng sein wie die in der übergeordneten Erteilung.

Diese Einschränkungen gelten nicht für Prinzipale, die die `CreateGrant`-Berechtigung aus einer Richtlinie erhalten, obwohl ihre Berechtigungen durch Richtlinienbedingungen eingeschränkt werden können.

Nehmen wir beispielsweise eine Erteilung, die es dem empfangenden Prinzipal ermöglicht, die `GenerateDataKey`-, `Decrypt`- und `CreateGrant`-Operationen aufzurufen. Wir nennen eine Erteilung, die die `CreateGrant`-Berechtigung erlaubt, eine übergeordnete Erteilung.

```
# 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"
                }
            },
        }
    ]
}
```

Der Empfänger-Prinzipal exampleUser kann diese Berechtigung verwenden, um eine Erteilung zu erstellen, die eine Teilmenge der in der ursprünglichen Erteilung angegebenen Operationen enthält – z. B. `CreateGrant` und `Decrypt`. Die untergeordnete Erteilung kann keine anderen Operationen enthalten (z. B. `ScheduleKeyDeletion` oder `ReEncrypt`).

Außerdem müssen die Erteilungs-Beschränkungen in untergeordneten Erteilungen mindestens ebenso restriktiv sein, wie die in der übergeordneten Erteilungen. So kann die untergeordnete Erteilung beispielsweise Paare zu einer `EncryptionContextSubset`-Beschränkung in der übergeordneten Erteilung hinzufügen. Sie kann sie jedoch nicht entfernen. Die untergeordnete Erteilung kann eine `EncryptionContextSubset`-Beschränkung in eine `EncryptionContextEquals`-Beschränkung ändern – nicht jedoch umgekehrt.

Beispielsweise kann der Empfänger-Prinzipal die `CreateGrant`-Berechtigung verwenden, die er von der übergeordneten Erteilung erhalten hat, um die folgende untergeordnete Erteilung zu erstellen Die Operationen in der untergeordneten Erteilung sind eine Teilmenge der Operationen in der übergeordneten Erteilung, und die Erteilungs-Beschränkungen sind restriktiver.

```
# 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": {
IAM best practices discourage the use of IAM users with long-term credentials. Whenever
                      possible, use IAM roles, which provide temporary credentials. For
 details,
                      see Security best practices in IAM in the IAM User Guide.
        "EncryptionContextEquals": {
                    "Department": "IT"
                }
```

```
            },
        }
    ]
}
```

Der Empfänger-Prinzipal in der untergeordneten Erteilung, `anotherUser`, kann seine `CreateGrant`-Berechtigung zum Erstellen von Erteilungen verwenden. Allerdings müssen die Erteilungen, die `anotherUser` erstellt, die Operationen in der übergeordneten Erteilung oder in einer Teilmenge enthalten, und die Erteilungs-Einschränkungen müssen gleichwertig oder strenger sein.

Verwalten von Erteilungen

Prinzipale mit den erforderlichen Berechtigungen können Erteilungen anzeigen, verwenden und löschen (aufheben oder widerrufen). Um die Berechtigungen zum Erstellen und Verwalten von Erteilungen zu verfeinern, unterstützt AWS KMS mehrere Richtlinienbedingungen, die Sie in Schlüsselrichtlinien und IAM-Richtlinien verwenden können.

Themen

- Steuerung des Zugriffs auf Erteilungen
- Anzeigen einer Erteilung
- Verwenden eines Erteilungs-Token
- Außerbetriebnahme und Widerruf von Erteilungen

Steuerung des Zugriffs auf Erteilungen

Sie können den Zugriff auf Operationen steuern, die Erteilungen in Schlüsselrichtlinien, IAM-Richtlinien und anderen Erteilungen erstellen und verwalten. Prinzipale, die die `CreateGrant`-Berechtigung von einer Erteilung erhalten, haben mehr eingeschränkte Erteilungs-Berechtigungen.

API-Produktion	Schlüsselrichtlinie oder IAM-Richtlinie	Gewährung
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-

API-Produktion	Schlüsselrichtlinie oder IAM-Richtlinie	Gewährung
Außerbetriebnahme einer Erteilung	(Begrenzt. Siehe Außerbetriebnahme und Widerruf von Erteilungen)	✓
RevokeGrant	✓	-

Wenn Sie eine Schlüsselrichtlinie oder eine IAM-Richtlinie verwenden, um den Zugriff auf Operationen zu steuern, die Erteilungen erstellen und verwalten, können Sie eine oder mehrere Richtlinienbedingungen verwenden, um die Berechtigung einzuschränken. AWS KMS unterstützt alle der folgenden Bedingungsschlüssel für Erteilungen. Ausführliche Informationen und Beispiele finden Sie unter AWS KMS Bedingungsschlüssel.

kms:GrantConstraintType

Erlaubt es Prinzipalen, eine Erteilung nur zu erstellen, wenn die Erteilung die angegebene Erteilungs-Einschränkung enthält.

kms:GrantIsForAWSResource

Erlaubt es Prinzipalen, `CreateGrant`, `ListGrants` oder `RevokeGrant` nur dann aufzurufen, wenn ein AWS-Service, der in AWS KMS integriert ist, die Anforderung im Namen des Prinzipals sendet.

kms:GrantOperations

Erlaubt es Prinzipalen, eine Erteilung zu erstellen, beschränkt jedoch die Erteilung auf die angegebenen Operationen.

kms:GranteePrincipal

Erlaubt es Prinzipalen, eine Erteilung nur für den angegebenen Empfänger-Prinzipal zu erstellen.

kms:RetiringPrincipal

Erlaubt es Prinzipalen, eine Erteilung nur zu erstellen, wenn die Erteilung einen bestimmten ausscheidenden Prinzipal angibt.

Anzeigen einer Erteilung

Um die Erteilung anzuzeigen, verwenden Sie die -ListGrantsOperation. Sie müssen den KMS-Schlüssel angeben, für den die Erteilungen gelten. Sie können die Liste der Erteilungen auch nach Erteilungs-ID oder Empfänger-Prinzipal filtern. Weitere Beispiele finden Sie unter Anzeigen einer Erteilung.

Um alle Erteilungen in der Region AWS-Konto und mit einem bestimmten ausscheidenden Prinzipal anzuzeigen, verwenden Sie ListRetirableGrants. Die Antworten enthalten Details zu den einzelnen Erteilungen.

> **Note**
>
> Das `GranteePrincipal`-Feld in der `ListGrants`-Antwort enthält normalerweise den Berechtigungsprinzipal der Genehmigung. Wenn der Empfänger-Prinzipal in der Erteilung jedoch ein AWS-Service ist, enthält das `GranteePrincipal`-Feld den Service-Prinzipal, der mehrere verschiedene Empfänger-Prinzipale repräsentieren kann.

Mit dem folgenden Befehl werden beispielsweise alle Erteilungen für einen KMS-Schlüssel aufgeführt.

```
$  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"
            ]
        }
    ]
}
```

Verwenden eines Erteilungs-Token

Die AWS KMS-API folgt einem eventuellen Konsistenzmodell. Beim Erstellen einer Erteilung ist die Erteilung möglicherweise nicht sofort gültig. Es kann zu einer kurzen Verzögerung kommen, bevor die Änderung in allen Bereichen von AWS KMS verfügbar ist. In der Regel dauert es weniger als ein paar Sekunden, bis sich die Änderung im gesamten System verbreitet, in einigen Fällen kann es jedoch mehrere Minuten dauern. Sobald sich die Änderung vollständig im System verbreitet hat, kann der Empfänger-Prinzipal die Berechtigungen in der Berechtigungserteilung verwenden, ohne das Berechtigungserteilungs-Token oder irgendwelche Beweise für die Berechtigungserteilung anzugeben. Wenn jedoch eine Erteilung so neu ist, dass sie noch nicht in allen Bereichen von AWS KMS bekannt ist, schlägt die Anforderung möglicherweise mit einem `AccessDeniedException`-Fehler fehl.

Um die Berechtigungen in einer neuen Erteilung sofort zu verwenden, verwenden Sie den Erteilungs-Token für die Erteilung. Speichern Sie das Erteilungs-Token, das der CreateGrant Vorgang zurückgibt. Senden Sie dann das Erteilungs-Token in der Anforderung für die AWS KMS-Produktion ab. Sie können ein Erteilungs-Token an jede AWS KMS-Erteilungs-Produktion senden und Sie können mehrere Erteilungs-Token in derselben Anforderung senden.

Im folgenden Beispiel wird die -`CreateGrant`Operation verwendet, um eine Erteilung zu erstellen, die die - GenerateDataKey und Decrypt-Operationen zulässt. Sie speichert das Erteilungs-Token, das `CreateGrant` in der `token`-Variable zurückgibt. Dann verwendet sie in einem Aufruf an die `GenerateDataKey`-Produktion das Erteilungs-Token in der `token`-Variable.

```
# 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
```

Prinzipale mit Berechtigung können auch ein Erteilungs-Token verwenden, um eine neue Erteilung aufzuheben, noch bevor die Erteilung über AWS KMS verfügbar ist. (Die `RevokeGrant`-Produktion akzeptiert kein Erteilungs-Token.) Details hierzu finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Außerbetriebnahme und Widerruf von Erteilungen

Um eine Erteilung zu löschen, müssen Sie sie aufheben oder widerrufen.

Die RevokeGrant Operationen RetireGrant und sind sehr ähnlich. Beide Operationen löschen eine Erteilung, wodurch die Berechtigungen, die die Erteilung erlaubt, eliminiert werden. Der Hauptunterschied zwischen diesen Operationen besteht darin, wie sie autorisiert werden.

RevokeGrant

: Wie die meisten AWS KMS-Operationen, wird der Zugriff auf die `RevokeGrant`-Produktion über Schlüsselrichtlinien und IAM-Richtlinien gesteuert. Die RevokeGrant API kann von jedem Prinzipal mit -`kms:RevokeGrant`Berechtigung aufgerufen werden. Diese Berechtigung ist in den Standard-Berechtigungen enthalten, die Schlüsseladministratoren erteilt werden. In der Regel widerrufen Administratoren eine Erteilung, um Berechtigungen zu verweigern, die die Erteilung erlaubt.

RetireGrant

: Die Erteilung bestimmt, wer sie aufheben kann. Mit diesem Entwurf können Sie den Lebenszyklus einer Erteilung steuern, ohne Schlüsselrichtlinien oder IAM-Richtlinien zu ändern. In der Regel können Sie eine Erteilung aufheben, wenn Sie die Berechtigungen nicht mehr benötigen.

  Eine Erteilung kann durch einen optionalen ausscheidenden Prinzipal, der in der Erteilung angegeben ist, aufgehoben werden. Der Empfänger-Prinzipal kann die Erteilung auch aufheben, aber nur, wenn er auch ein ausscheidender Prinzipal ist oder die Erteilung die `RetireGrant`-Produktion enthält. Als Backup kann das AWS-Konto, in dem die Erteilung erstellt wurde, die Erteilung aufheben.

Es gibt eine `kms:RetireGrant`-Berechtigung, die in IAM-Richtlinien verwendet werden kann, aber sie verfügt über eingeschränkte Funktionalität. Prinzipale, die in der Erteilung angegeben sind, können eine Erteilung ohne die `kms:RetireGrant`-Berechtigung aufheben. Die `kms:RetireGrant`-Berechtigung allein erlaubt es Prinzipalen nicht, eine Erteilung aufzuheben. Die `kms:RetireGrant`-Berechtigung ist in einer Schlüsselrichtlinie nicht wirksam.

- Um die Berechtigung zur Außerbetriebnahme einer Erteilung zu verweigern, können Sie eine `Deny`-Aktion mit der `kms:RetireGrant`-Berechtigung verwenden.
- Das AWS-Konto, das der Eigentümer des KMS-Schlüssels ist, kann die `kms:RetireGrant`-Berechtigung an einen IAM-Prinzipal im Konto delegieren.
- Wenn der ausscheidende Prinzipal ein anderes AWS-Konto ist, können Administratoren in dem anderen Konto `kms:RetireGrant` verwenden, um die Berechtigung zur Außerbetriebnahme der Erteilung an einen IAM-Prinzipal in diesem Konto zu delegieren.

Die AWS KMS-API folgt einem eventuellen Konsistenzmodell. Wenn Sie eine Erteilung erstellen, aufheben oder widerrufen, kann es zu einer kurzen Verzögerung kommen, bevor die Änderung in allen Bereichen von AWS KMS verfügbar ist. In der Regel dauert es weniger als ein paar Sekunden, bis sich die Änderung im gesamten System verbreitet, in einigen Fällen kann es jedoch mehrere Minuten dauern. Wenn Sie eine neue Erteilung sofort löschen müssen, bevor sie in AWS KMS verfügbar ist, Verwenden Sie einen Erteilungs-Token, um die Erteilung aufzuheben. Sie können kein Erteilungs-Token verwenden, um eine Erteilung zu widerrufen.

Verbindung zu AWS KMS über einen VPC-Endpunkt

Sie können über einen privaten Schnittstellenendpunkt in Ihrer Virtual Private Cloud (VPC) eine direkte Verbindung zu AWS KMS herzustellen. Wenn Sie den VPC-Endpunkt einer Schnittstelle verwenden, findet die Kommunikation zwischen Ihrer VPC und AWS KMS vollständig innerhalb des AWS-Netzwerks statt.

AWS KMS unterstützt Endpunkte von Amazon Virtual Private Cloud (Amazon VPC), die von AWS PrivateLink bereitgestellt werden. Jeder VPC-Endpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen (ENIs) mit privaten IP-Adressen in Ihren VPC-Subnetzen repräsentiert.

Der Schnittstellen-VPC-Endpunkt verbindet Ihre VPC direkt mit AWS KMS, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung. Die Instances in Ihrer VPC benötigen für die Kommunikation mit AWS KMS keine öffentlichen IP-Adressen.

Regionen

AWS KMS unterstützt VPC-Endpunkte und VPC-Endpunktrichtlinien in allen AWS-Regionen, in denen AWS KMS unterstützt wird.

Themen

Überlegungen zu AWS KMS-VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für AWS KMS einrichten, lesen Sie unbedingt das Thema Schnittstellen-Endpunkte, Eigenschaften und Einschränkungen im AWS PrivateLink-Leitfaden.

AWS KMS-Support für einen VPC-Endpunkt umfasst Folgendes.

- Sie können den VPC-Endpunkt verwenden, um alle AWS KMS-API-Vorgänge in Ihrer VPC aufzurufen.
- Sie können einen Schnittstellen-VPC-Endpunkt erstellen, der eine Verbindung mit einem AWS KMS-Regionsendpunkt oder einem AWS KMS-FIPS-Endpunkt herstellt.
- Sie können anhand von AWS CloudTrail-Protokollen Ihre Verwendung von KMS-Schlüsseln durch den VPC-Endpunkt überwachen. Details hierzu finden Sie unter Protokollieren des VPC-Endpunkts.

Erstellung eines VPC-Endpunkts für AWS KMS

Sie können einen VPC-Endpunkt für AWS KMS mithilfe der Amazon-VPC-Konsole oder der Amazon-VPC-API erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink-Leitfaden.

- Zur Erstellung eines VPC-Endpunkts für AWS KMS verwenden Sie den folgenden Servicenamen:

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

In der Region USA West (Oregon) (`us-west-2`) würde der Servicename wie folgt lauten:

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

- Verwenden Sie den folgenden Servicenamen, um einen VPC-Endpunkt zu erstellen, der eine Verbindung zu einem AWS KMS-FIPS-Endpunkt herstellt:

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

In der Region USA West (Oregon) (`us-west-2`) würde der Servicename wie folgt lauten:

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

Um die Verwendung des VPC-Endpunkts zu vereinfachen, können Sie einen privaten DNS-Namen für Ihren VPC-Endpunkt aktivieren. Wenn Sie die Option Enable Private DNS Name (Privaten DNS-Namen aktivieren) auswählen, wird der standardmäßige AWS KMS-DNS-Hostname in Ihren VPC-Endpunkt aufgelöst. `https://kms.us-west-2.amazonaws.com` würde beispielsweise in einen VPC-Endpunkt aufgelöst, der mit dem Servicenamen `com.amazonaws.us-west-2.kms` verbunden ist.

Diese Option vereinfacht die Verwendung des VPC-Endpunkts. Die AWS und die AWS CLI-SDKs verwenden darüber hinaus standardmäßig den AWS KMS-Standard-DNS-Hostnamen. Daher müssen Sie die URL des VPC-Endpunkts in Anwendungen und Befehlen nicht angeben.

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im AWS PrivateLink-Leitfaden.

Herstellen einer Verbindung mit einem AWS KMS-VPC-Endpunkt

Sie können über einen AWS-SDK, über die AWS CLI oder über AWS Tools for PowerShell mittels eines VPC-Endpunkts eine Verbindung mit AWS KMS herstellen. Um den VPC-Endpunkt anzugeben, verwenden Sie seinen DNS-Namen.

Dieser list-keys-Befehl verwendet zur Angabe des VPC-Endpunkts z. B. den Parameter `endpoint-url`. Wenn Sie einen solchen Befehl verwenden möchten, ersetzen Sie die Beispiels-ID des VPC-Endpunkts durch eine ID in Ihrem Konto.

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

Wenn Sie beim Erstellen Ihres VPC-Endpunkts private Hostnamen aktiviert waren, müssen Sie die URL des Endpunkts in Ihren CLI-Befehlen oder in Ihrer Anwendungskonfiguration angeben. In diesem Fall wird der standardmäßige AWS KMS-DNS-Hostname in Ihren VPC-Endpunkt aufgelöst. Die AWS CLI und SDKs verwenden diesen Hostnamen standardmäßig. Daher sind vor der Verwendung des VPC-Endpunkts zum Herstellen einer Verbindung mit einem regionalen AWS KMS-Endpunkt keine Änderungen in Ihren Skripts und in Ihrer Anwendung erforderlich.

Zur Verwendung privater Hostnamen müssen die Attribute `enableDnsHostnames` und `enableDnsSupport` Ihrer VPC auf `true` eingestellt sein. Um diese Attribute festzulegen, verwenden Sie die -ModifyVpcAttributeOperation. Details dazu finden Sie unter Anzeigen und Aktualisieren von DNS-Attributen für Ihre VPC im Amazon-VPC-Benutzerhandbuch.

Steuern des Zugriffs auf einen VPC-Endpunkt

Um den Zugriff auf Ihren VPC-Endpunkt für AWS KMS zu steuern, fügen Sie Ihrem VPC-Endpunkt eine VPC Endpunktrichtlinie an. Die Endpunktrichtlinie bestimmt, ob Prinzipale den VPC-Endpunkt verwenden können, um AWS KMS-Operationen für AWS KMS-Ressourcen aufzurufen.

Sie können beim Erstellen des Endpunkts eine VPC-Endpunktrichtlinie erstellen und die VPC-Endpunktrichtlinie jederzeit ändern. Verwenden Sie die VPC-Managementkonsole oder die -CreateVpcEndpoint oder -ModifyVpcEndpointOperationen. Sie können eine VPC-Endpunktrichtlinie auch erstellen und ändern, indem Sie eine AWS CloudFormation-Vorlage verwenden. Hilfe zur Verwendung der VPC-Managementkonsole finden Sie unter Erstellen eines Schnittstellenendpunkts und Ändern eines Schnittstellenendpunkts im AWS PrivateLink-Leitfaden.

> **Note**
>
> AWS KMS unterstützt VPC-Endpunktrichtlinien ab Juli 2020. VPC-Endpunkte für AWS KMS, die vor diesem Datum erstellt wurden, haben die Standard-VPC-Endpunktrichtlinie. Sie können diese jedoch jederzeit ändern.

Hilfe beim Schreiben und Formatieren eines JSON-Richtliniendokuments finden Sie in der IAM-JSON-Richtlinienreferenz im IAM-Benutzerhandbuch.

Themen

Weitere Informationen über VPC-Endpunktrichtlinien

Damit eine AWS KMS-Anforderung, die einen VPC-Endpunkt verwendet, erfolgreich ist, benötigt der Prinzipal Berechtigungen aus zwei Quellen:

- Eine -Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung muss dem Prinzipal die Berechtigung erteilen, die Operation für die Ressource (KMS-Schlüssel oder Alias) aufzurufen.
- Eine VPC-Endpunktrichtlinie muss dem Prinzipal die Berechtigung erteilen, den Endpunkt für die Anforderung zu verwenden.

So kann eine Schlüsselrichtlinie einem Prinzipal die Berechtigung zum Aufrufen von Decrypt auf einen bestimmten KMS-Schlüssel erteilen. Allerdings könnte die VPC-Endpunktrichtlinie diesem Prinzipal nicht erlauben, `Decrypt` auf diesem KMS-Schlüssel mithilfe des Endpunkts aufzurufen.

Oder eine VPC-Endpunktrichtlinie kann es einem Prinzipal ermöglichen, den Endpunkt für den Aufruf DisableKey bestimmter KMS-Schlüssel zu verwenden. Wenn der Prinzipal jedoch nicht über diese Berechtigungen aus einer Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung verfügt, schlägt die Anforderung fehl.

Standard-VPC-Endpunktrichtlinie

Jeder VPC-Endpunkt verfügt über eine VPC-Endpunktrichtlinie. Sie müssen die Richtlinie jedoch nicht angeben. Wenn Sie keine Richtlinie angeben, erlaubt die standardmäßige Endpunktrichtlinie alle Operationen aller Prinzipale auf allen Ressourcen über den Endpunkt.

Allerdings muss der Prinzipal für AWS KMS-Ressourcen auch die Berechtigung zum Aufrufen der Operation von einer Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung haben. Daher besagt die

Standardrichtlinie in der Praxis, dass, wenn ein Prinzipal über die Berechtigung zum Aufrufen einer Operation für eine Ressource verfügt, diese auch mithilfe des Endpunkts aufrufen kann.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Damit Prinzipale den VPC-Endpunkt nur für eine Teilmenge ihrer zulässigen Operationen verwenden können, erstellen oder aktualisieren Sie die VPC-Endpunktrichtlinie.

Erstellen einer VPC-Endpunktrichtlinie

Eine VPC-Endpunktrichtlinie bestimmt, ob ein Prinzipal die Berechtigung hat, den VPC-Endpunkt zum Ausführen von Operationen auf einer Ressource zu verwenden. Allerdings muss der Prinzipal für AWS KMS-Ressourcen auch die Berechtigung zum Aufrufen der Operation von einer Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung haben.

Für jede VPC-Endpunktrichtlinie sind die folgenden Elemente erforderlich:

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

Die Richtlinienanweisung gibt den VPC-Endpunkt nicht an. Stattdessen gilt sie für jeden VPC-Endpunkt, dem die Richtlinie angefügt ist. Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon-VPC-Benutzerhandbuch.

Das Folgende ist ein Beispiel für eine VPC-Endpunktrichtlinie für AWS KMS. An einen VPC-Endpunkt angefügt, erlaubt diese Richtlinie es `ExampleUser`, den VPC-Endpunkt zum Aufrufen der angegebenen Operationen für die angegebenen KMS-Schlüssel zu verwenden. Bevor Sie eine solche Richtlinie verwenden, ersetzen Sie den Beispiel-Prinzipal und Schüssel-ARN mit gültigen Werten aus Ihrem Konto.

```
{
   "Statement":[
      {
         "Sid": "AllowDecryptAndView",
         "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
         "Effect":"Allow",
         "Action": [
             "kms:Decrypt",
             "kms:DescribeKey",
             "kms:ListAliases",
             "kms:ListKeys"
          ],
         "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
   ]
}
```

AWS CloudTrail protokolliert alle Operationen, die den VPC-Endpunkt verwenden. Ihre CloudTrail Protokolle enthalten jedoch keine Operationen, die von Prinzipalen in anderen Konten angefordert werden, oder Operationen für KMS-Schlüssel in anderen Konten.

Daher möchten Sie möglicherweise eine VPC-Endpunktrichtlinie erstellen, die verhindert, dass Prinzipale in externen Konten den VPC-Endpunkt zum Aufrufen von AWS KMS-Operationen für alle Schlüssel im lokalen Konto nutzen.

Im folgenden Beispiel wird der globale Bedingungsschlüssel aws:PrincipalAccount verwendet, um allen Prinzipalen den Zugriff für alle Operationen auf allen KMS-Schlüsseln zu verweigern, es sei denn, der Prinzipal befindet sich im lokalen Konto. Bevor Sie eine Richtlinie wie diese verwenden, ersetzen Sie die Beispiel-Konto-ID durch eine gültige.

```
{
  "Statement": [
    {
      "Sid": "AccessForASpecificAccount",
      "Principal": {"AWS": "*"},
      "Action": "kms:*",
      "Effect": "Deny",
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "111122223333"
```

```
        }
      }
    }
  ]
}
```

Anzeigen einer VPC-Endpunktrichtlinie

Um die VPC-Endpunktrichtlinie für einen Endpunkt anzuzeigen, verwenden Sie die VPC-Managementkonsole oder die -DescribeVpcEndpointsOperation.

Der folgende AWS CLI-Befehl ruft die Richtlinie für den Endpunkt mit der angegebenen VPC-Endpunkt-ID ab.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispiel-Endpunkt-ID durch eine gültige aus Ihrem Konto.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

Verwenden eines VPC-Endpunkts in einer Richtlinienanweisung

Sie können den Zugriff auf AWS KMS-Ressourcen und -Operationen steuern, wenn die Anforderung von VPC stammt oder einen VPC-Endpunkt verwendet. Verwenden Sie dazu einen der folgenden globalen Bedingungsschlüssel in einer Schlüsselrichtlinie oder IAM-Richtlinie.

- Verwenden Sie den `aws:sourceVpce`-Bedingungsschlüssel zum Erteilen oder Beschränken des Zugriffs anhand des VPC-Endpunkts.
- Verwenden Sie den `aws:sourceVpc`-Bedingungsschlüssel zum Erteilen oder Beschränken des Zugriffs anhand des VPC, auf der der private Endpunkt gehostet wird.

> **Note**
>
> Beim Erstellen von Schlüsselrichtlinien und IAM-Richtlinien anhand von Ihrem VPC-Endpunkt ist Vorsicht geboten. Wenn eine Richtlinienanweisung verlangt, dass Anforderungen von einer bestimmten VPC oder einem bestimmten VPC-Endpunkt stammen müssen, schlagen Anforderungen von integrierten AWS-Services, die eine AWS KMS-Ressource in Ihrem

Namen verwenden, möglicherweise fehl. Weitere Informationen dazu finden Sie unter Verwenden von VPC-Endpunkt-Bedingungen in Richtlinien mit AWS KMS -Berechtigungen. Weiterhin ist der Bedingungsschlüssel `aws:sourceIP` nicht wirksam, wenn die Anforderung von einem Amazon-VPC-Endpunkt kommt. Um die Anforderungen an einen VPC-Endpunkt zu beschränken, verwenden Sie die Bedingungsschlüssel `aws:sourceVpce` oder `aws:sourceVpc`. Weitere Informationen finden Sie unter Identity and Access Management für VPC-Endpunkte und VPC-Endpunkt-Services im AWS PrivateLink-Leitfaden.

Sie können diese globalen Bedingungsschlüssel verwenden, um den Zugriff auf AWS KMS keys (KMS-Schlüssel), Aliase und Operationen wie zu steuern, CreateKey die nicht von einer bestimmten Ressource abhängen.

Die folgende Beispiel-Schlüsselrichtlinie erlaubt es einem Benutzer z. B., nur dann kryptografische Operationen mit einem KMS-Schlüssel durchzuführen, wenn die Anforderung den angegebenen VPC-Endpunkt nutzt. Wenn ein Benutzer eine Anforderung an AWS KMS ausgibt, wird die VPC-Endpunkt-ID in der Anforderung mit dem `aws:sourceVpce`-Bedingungsschlüsselwert in der Richtlinie verglichen. Wenn sie nicht übereinstimmen, wird die Anforderung abgelehnt.

Um eine Richtlinie wie diese zu verwenden, ersetzen Sie die Platzhalter-ID des AWS-Konto und VPC-Endpunkt-IDs durch gültige Werte für Ihr Konto.

```
{
    "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"
                }
            }
        }

    ]
}
```

Sie können auch den `aws:sourceVpc`-Bedingungsschlüssel verwenden, um den Zugriff auf Ihre KMS-Schlüssel anhand der VPC, in der sich der VPC-Endpunkt befindet, zu beschränken.

Die folgende Beispiel-Schlüsselrichtlinie erlaubt nur dann Befehle zur Verwaltung des KMS-Schlüssels, wenn sie von `vpc-12345678` stammen. Außerdem erlaubt sie nur Befehle, die den KMS-Schlüssel für kryptographische Operationen verwenden, wenn sie von `vpc-2b2b2b2b` stammen. Sie verwenden eine solche Richtlinie wie diese möglicherweise, wenn eine Anwendung in einer VPC ausgeführt wird, aber eine zweite, isolierte VPC für die Verwaltungsfunktionen genutzt wird.

Um eine Richtlinie wie diese zu verwenden, ersetzen Sie die Platzhalter-ID des AWS-Konto und VPC-Endpunkt-IDs durch gültige Werte für Ihr Konto.

```
{
    "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": "*",
        }
    ]
}
```

Protokollieren des VPC-Endpunkts

AWS CloudTrail protokolliert alle Operationen, die den VPC-Endpunkt verwenden. Wenn eine Anforderung an AWS KMS einen VPC-Endpunkt verwendet, wird die VPC-Endpunkt-ID in dem AWS CloudTrail-Protokoll-Eintrag angezeigt, mit dem die Anforderung aufgezeichnet wird. Sie können mit der Endpunkt-ID die Verwendung Ihres AWS KMS-VPC-Endpunkts überwachen.

Ihre CloudTrail Protokolle enthalten jedoch keine Operationen, die von Prinzipalen in anderen Konten angefordert werden, oder Anforderungen für AWS KMS Operationen an KMS-Schlüsseln und Aliassen in anderen Konten. Um Ihre VPC zu schützen, werden Anforderungen, die von einer VPC-Endpunktrichtlinie verweigert werden, aber ansonsten erlaubt gewesen wären, nicht in AWS CloudTrail erfasst.

Dieser Beispiel-Protokolleintrag zeichnet z. B. eine GenerateDataKey-Anforderung auf, die den VPC-Endpunkt genutzt hat. Das Feld `vpcEndpointId` erscheint am Ende des Protokolleintrags.

```
{
  "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"
}
```

Zustandstasten für AWS KMS

Sie können Bedingungen in den Schlüsselrichtlinien und IAM-Richtlinien angeben, die den Zugriff auf AWS KMS Ressourcen steuern. Die Richtlinienanweisung ist nur wirksam, wenn diese Bedingungen erfüllt sind. Beispielsweise kann festgelegt werden, dass eine Richtlinienanweisung erst ab einem bestimmten Datum gilt. Sie können auch festlegen, dass eine Richtlinienanweisung den Zugriff nur steuert, wenn in einer API-Anforderung ein bestimmter Wert vorhanden ist.

Verwenden Sie Bedingungsschlüssel im `Condition`-Element einer Richtlinienanweisung mit IAM-Bedingungs-Operatoren, um Bedingungen festzulegen. Einige Bedingungsschlüssel gelten allgemein für AWS, andere sind spezifisch für. AWS KMS

Bedingungsschlüsselwerte müssen den Zeichen- und Kodierungsregeln für AWS KMS wichtige Richtlinien und IAM-Richtlinien entsprechen. Weitere Informationen zu wichtigen Dokumentenregeln für Schlüsselrichtlinien finden Sie unter Schlüsselrichtlinienformat. Weitere Informationen zu Regeln für IAM-Richtliniendokumente finden Sie unter Anforderungen für den IAM-Namen im IAM-Benutzerhandbuch.

Themen

- AWS globale Bedingungsschlüssel
- AWS KMS Bedingungsschlüssel
- AWS KMS Bedingungsschlüssel für AWS Nitro Enclaves

AWS globale Bedingungsschlüssel

AWS definiert globale Bedingungsschlüssel, eine Reihe von Schlüsseln für Richtlinienbedingungen für alle AWS Dienste, die IAM für die Zugriffskontrolle verwenden. AWS KMS unterstützt alle globalen Bedingungsschlüssel. Sie können sie in AWS KMS wichtigen Richtlinien und IAM-Richtlinien verwenden.

Sie können beispielsweise den PrincipalArn globalen Bedingungsschlüssel aws: verwenden, um den Zugriff auf einen AWS KMS key (KMS-Schlüssel) nur dann zuzulassen, wenn der Principal in der Anfrage durch den Amazon-Ressourcennamen (ARN) im Bedingungsschlüsselwert repräsentiert wird. Um die attributebasierte Zugriffskontrolle (ABAC) in zu unterstützen AWS KMS, können Sie den globalen Bedingungsschlüssel aws:ResourceTag/tag-key in einer IAM-Richtlinie verwenden, um den Zugriff auf KMS-Schlüssel mit einem bestimmten Tag zu ermöglichen.

Um zu verhindern, dass ein AWS Dienst in einer Richtlinie, in der der Principal ein AWS Dienstprinzipal ist, als verwirrter Stellvertreter verwendet wird, können Sie die globalen Bedingungsschlüssel oder verwenden. aws:SourceArnaws:SourceAccount Details hierzu finden Sie unter Verwenden der Bedingungsschlüssel `aws:SourceArn` oder `aws:SourceAccount`.

Informationen zu AWS globalen Bedingungsschlüsseln, einschließlich der Anforderungstypen, in denen sie verfügbar sind, finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch. Beispiele für die Verwendung globaler Bedingungsschlüssel in IAM-Richtlinien finden Sie unter Steuern des Zugriffs auf Anforderungen und Steuern von Tag-Schlüsseln im IAM-Benutzerhandbuch.

Die folgenden Themen bieten spezielle Anleitungen für die Verwendung von Bedingungsschlüsseln basierend auf IP-Adressen und VPC-Endpunkten.

Themen

- Verwenden der IP-Adressbedingung in Richtlinien mit AWS KMS -Berechtigungen
- Verwenden von VPC-Endpunkt-Bedingungen in Richtlinien mit AWS KMS -Berechtigungen

Verwenden der IP-Adressbedingung in Richtlinien mit AWS KMS -Berechtigungen

Sie können AWS KMS es verwenden, um Ihre Daten in einem integrierten AWS Service zu schützen. Seien Sie jedoch vorsichtig, wenn Sie die Operatoren für die `aws:SourceIp` IP-Adressbedingung oder den Bedingungsschlüssel in derselben Richtlinienerklärung angeben, die den Zugriff AWS KMS gewährt oder verweigert. Beispielsweise beschränkt die Richtlinie in AWS: Verweigert den Zugriff auf AWS Basierend auf der Quell-IP AWS Aktionen auf Anfragen aus dem angegebenen IP-Bereich.

Betrachten Sie folgendes Szenario:

1. Sie fügen einer IAM-Identität eine Richtlinie wie die unter AWS: Verweigert den Zugriff auf AWS Basierend auf der Quell-IP gezeigte Richtlinie an. Sie legen als Wert für den `aws:SourceIp`-Bedingungsschlüssel den IP-Adressbereich für das Unternehmen des Benutzers fest. Dieser IAM-Identität wurden andere Richtlinien angefügt, die es ihr erlauben, Amazon EBS, Amazon EC2 und AWS KMS zu verwenden.
2. Die Identität versucht, ein verschlüsseltes EBS-Volume an eine EC2-Instance anzufügen. Diese Aktion schlägt aufgrund eines Autorisierungsfehlers fehl, obwohl der Benutzer zur Verwendung aller relevanten Services berechtigt ist.

Schritt 2 schlägt fehl, weil die Anfrage AWS KMS zur Entschlüsselung des verschlüsselten Datenschlüssels des Volumes von einer IP-Adresse stammt, die mit der Amazon EC2 EC2-Infrastruktur verknüpft ist. Dieser Schritt wird nur erfolgreich durchgeführt, wenn die Anforderung von der IP-Adresse des ursprünglichen Benutzers stammt. Da die Richtlinie in Schritt 1 explizit alle Anforderungen von anderen als den angegebenen IP-Adressen ablehnt, wird die Berechtigung für Amazon EC2 zum Entschlüsseln des verschlüsselten Datenschlüssels des EBS-Datenträgers abgelehnt.

Weiterhin ist der Bedingungsschlüssel `aws:sourceIP` nicht wirksam, wenn die Anforderung von einem Amazon-VPC-Endpunkt kommt. Um Anforderungen an einen VPC-Endpunkt, einschließlich eines AWS KMS -VPC-Endpunkts zu beschränken, verwenden Sie die `aws:sourceVpce`- oder `aws:sourceVpc`-Bedingungsschlüssel. Weitere Informationen finden Sie unter VPC-Endpunkte – Steuern der Nutzung von Endpunkten im Amazon VPC-Benutzerhandbuch.

Verwenden von VPC-Endpunkt-Bedingungen in Richtlinien mit AWS KMS - Berechtigungen

AWS KMS unterstützt Amazon Virtual Private Cloud (Amazon VPC) -Endpunkte, die von betrieben werden. AWS PrivateLink Sie können die folgenden globalen Bedingungsschlüssel in wichtigen Richtlinien und IAM-Richtlinien verwenden, um den Zugriff auf AWS KMS Ressourcen zu steuern, wenn die Anfrage von einer VPC kommt oder einen VPC-Endpunkt verwendet. Details hierzu finden Sie unter Verwenden eines VPC-Endpunkts in einer Richtlinienanweisung.

- `aws:SourceVpc` beschränkt den Zugriff auf Anforderungen von der angegebenen VPC.
- `aws:SourceVpce` beschränkt den Zugriff auf Anforderungen vom angegebenen VPC-Endpunkt.

Wenn Sie diese Bedingungsschlüssel verwenden, um den Zugriff auf KMS-Schlüssel zu steuern, verweigern Sie möglicherweise versehentlich den Zugriff auf AWS Dienste, die in Ihrem Namen verwendet werden. AWS KMS

Seien Sie sorgsam darum bemüht, eine Situation wie das IP-Adressen-Bedingungsschlüssel Beispiel zu vermeiden. Wenn Sie Anfragen für einen KMS-Schlüssel auf einen VPC- oder VPC-Endpunkt beschränken, schlagen Aufrufe AWS KMS von einem integrierten Service wie Amazon S3 oder Amazon EBS möglicherweise fehl. Dies kann auch dann vorkommen, wenn die Quell-Anforderung letztendlich von der VPC oder dem VPC-Endpunkt stammt.

AWS KMS Bedingungsschlüssel

AWS KMS stellt eine Reihe von Bedingungsschlüsseln bereit, die Sie in wichtigen Richtlinien und IAM-Richtlinien verwenden können. Diese Bedingungsschlüssel sind spezifisch für. AWS KMS Sie können beispielsweise den Bedingungsschlüssel `kms:EncryptionContext:context-key` verwenden, damit ein bestimmter Verschlüsselungskontext bei der Steuerung des Zugriffs auf einen KMS-Schlüssel zur symmetrischen Verschlüsselung erforderlich ist.

Bedingungen für die Anforderung einer API-Produktion

Viele AWS KMS Bedingungsschlüssel steuern den Zugriff auf einen KMS-Schlüssel auf der Grundlage des Werts eines Parameters in der Anforderung für einen AWS KMS Vorgang. Sie können beispielsweise den KeySpec Bedingungsschlüssel kms: in einer IAM-Richtlinie verwenden, um die Verwendung des CreateKeyVorgangs nur dann zuzulassen, wenn der Wert des `KeySpec` Parameters in der `CreateKey` Anforderung lautet`RSA_4096`.

Dieser Bedingungstyp funktioniert sogar dann, wenn der Parameter nicht in der Anforderung angezeigt wird, z. B. wenn Sie den Standardwert des Parameters verwenden. Sie können beispielsweise den KeySpec Bedingungsschlüssel kms: verwenden, um Benutzern zu ermöglichen, die `CreateKey` Operation nur dann zu verwenden, wenn der Wert des `KeySpec` Parameters ist`SYMMETRIC_DEFAULT`, was der Standardwert ist. Diese Bedingung gewährt Anforderungen mit dem Parameter `KeySpec` und dem Wert `SYMMETRIC_DEFAULT` sowie Anforderungen ohne den Parameter `KeySpec`.

Bedingungen für KMS-Schlüssel, die in API-Operationen verwendet werden

Einige AWS KMS Bedingungsschlüssel können den Zugriff auf Operationen auf der Grundlage einer Eigenschaft des KMS-Schlüssels steuern, der für den Vorgang verwendet wird. Sie können beispielsweise die KeyOrigin Bedingung kms: verwenden, um es Prinzipalen zu ermöglichen, einen KMS-Schlüssel nur dann GenerateDataKeyaufzurufen, wenn `Origin` der KMS-Schlüssel aktiviert ist`AWS_KMS`. Um herauszufinden, ob ein Bedingungsschlüssel auf diese Weise verwendet werden kann, lesen Sie die Beschreibung des Bedingungsschlüssels.

Bei der Produktion muss es sich um eine KMS-Schlüsselressourcen-Produktion handeln, das heißt, eine Produktion, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Produktion. Wenn Sie diese Art von Bedingungsschlüssel beispielsweise für einen Vorgang verwenden, der für eine bestimmte KMS-Schlüsselressource nicht autorisiert ist ListKeys, ist die

Berechtigung nicht wirksam, da die Bedingung niemals erfüllt werden kann. Es ist keine KMS-Schlüssel-Ressource an der Autorisierung der Produktion `ListKeys` beteiligt und keine `KeySpec`-Eigenschaft.

In den folgenden Themen werden die einzelnen AWS KMS Bedingungsschlüssel beschrieben. Sie enthalten auch Beispiele für Richtlinienanweisungen, die die Richtliniensyntax veranschaulichen.

Verwenden von Satz-Operatoren mit Bedingungsschlüssel

Wenn eine Richtlinienbedingung zwei Gruppen von Werten vergleicht, z. B. die Gruppe von Tags in einer Anforderung und die Gruppe von Tags in einer Richtlinie, müssen Sie angeben, AWS wie die Gruppen verglichen werden sollen. IAM definiert für diesen Zweck zwei Satz-Operatoren, `ForAnyValue` und `ForAllValues`. Verwenden Sie Satz-Operatoren nur mit mehrwertigen Bedingungsschlüssel, die diese erfordern. Verwenden Sie keine Satz-Operatoren mit einzelwertigen Bedingungsschlüssel. Testen Sie Ihre Richtlinienanweisungen immer gründlich, bevor Sie sie in einer Operationsumgebung verwenden.

Bedingungsschlüssel sind einzelwertig oder mehrwertig. Informationen dazu, ob ein AWS KMS Bedingungsschlüssel ein- oder mehrwertig ist, finden Sie in der Spalte Wertetyp in der Beschreibung des Bedingungsschlüssels.

- Einzelwertige Bedingungsschlüssel haben höchstens einen Wert im Autorisierungskontext (die Anforderung oder Ressource). Da beispielsweise jeder API-Aufruf nur von einem API-Aufruf ausgehen kann AWS-Konto, CallerAccount ist kms: ein einwertiger Bedingungsschlüssel. Verwenden Sie keinen Satz-Operator mit einem einzelwertigen Bedingungsschlüssel.
- Mehrwertige Bedingungsschlüssel haben mehrfache Wert im Autorisierungskontext (die Anforderung oder Ressource). Da beispielsweise jeder KMS-Schlüssel mehrere Aliase haben kann, ResourceAliases kann kms: mehrere Werte haben. Mehrwertige Bedingungsschlüssel erfordern einen Satz-Operator.

Beachten Sie, dass der Unterschied zwischen einzelwertigen und mehrwertigen Bedingungsschlüsseln von der Anzahl der Werte im Autorisierungskontext abhängt, nicht von der Anzahl der Werte in der Richtlinienbedingung.

> **Warning**
>
> Wenn Sie einen Satz-Operator mit einem einzelwertigen Bedingungsschlüssel verwenden, können Sie eine Richtlinienanweisung erstellen, die übermäßig permissiv (oder zu restriktiv)

ist. Verwenden Sie Satz-Operatoren nur mit mehrwertigen Bedingungsschlüssel, die diese erfordern.
Wenn Sie eine Richtlinie erstellen oder aktualisieren, die einen `ForAllValues` Set-Operator mit dem Kontextschlüssel kms:EncryptionContext: oder den `aws:RequestTag/`*`tag-key`* Bedingungsschlüsseln enthält, wird die folgende AWS KMS Fehlermeldung zurückgegeben: `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.`

Ausführliche Informationen zu den `ForAnyValue`- und `ForAllValues`-Satz-Operatoren finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch. Informationen zum Risiko der Verwendung des `ForAllValues` Set-Operators mit einer einwertigen Bedingung finden Sie unter Sicherheitswarnung — ForAllValues mit einwertigem Schlüssel im IAM-Benutzerhandbuch.

Themen

- km: MacAlgorithm
- km: MessageType
- km: MultiRegion
- km: MultiRegionKeyType
- km: PrimaryRegion
- km: ReEncryptOnSameKey
- km: RequestAlias
- km: ResourceAliases
- km: ReplicaRegion
- km: RetiringPrincipal
- km: RotationPeriodInDays
- km: ScheduleKeyDeletionPendingWindowInDays
- km: SigningAlgorithm
- km: ValidTo
- km: ViaService
- km: WrappingAlgorithm
- km: WrappingKeySpec

km: BypassPolicyLockoutSafetyCheck

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:BypassPolicyLockoutSafetyCheck`	Boolesch	Einzelwertig	`CreateKey` `PutKeyPolicy`	Nur IAM-Richtlinien Schlüsselrichtlinien und IAM-Richtlinien

Der `kms:BypassPolicyLockoutSafetyCheck` Bedingungsschlüssel steuert den Zugriff auf die PutKeyPolicyOperationen CreateKeyund auf der Grundlage des Werts des `BypassPolicyLockoutSafetyCheck` Parameters in der Anforderung.

Die folgende Beispiel-IAM-Richtlinienanweisung verhindert, dass Benutzer die Richtliniensperre-Sicherheitsprüfung umgehen, indem Sie die Berechtigung zum Erstellen von KMS-Schlüssel ablehnen, wenn der Wert des Parameters `BypassPolicyLockoutSafetyCheck` in der `CreateKey`-Anforderung `true.` lautet.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Außerdem können Sie den `kms:BypassPolicyLockoutSafetyCheck`-Bedingungsschlüssel in einer IAM- oder Schlüsselrichtlinie verwenden, um den Zugriff auf die `PutKeyPolicy`-Produktion zu steuern. Die folgende Beispiel-Richtlinienanweisung einer Schlüsselrichtlinie verhindert, dass Benutzer beim Ändern der Richtlinie eines KMS-Schlüssel die Richtliniensperre-Sicherheitsprüfung umgehen.

Statt eine explizite `Deny`-Bedingung zu verwenden, nutzt diese Richtlinienanweisung `Allow` mit dem Null-bedingten Operator, um den Zugriff nur zu erlauben, wenn die Anforderung keinen `BypassPolicyLockoutSafetyCheck`-Parameter enthält. Wenn der Parameter nicht verwendet wird, lautet der Standardwert `false`. Diese etwas schwächere Richtlinienanweisung kann überschrieben werden, wenn ein Bypass erforderlich ist. Dies ist allerdings selten der Fall.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
```

```
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Informationen finden Sie auch unter:

- km: KeySpec
- km: KeyOrigin
- km: KeyUsage

km: CallerAccount

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:CallerAccount`	String	Einzelwertig	KMS-Schlüsselressourcen-Operationen Benutzerdefinierter-Schlüsselspeicher-Operationen	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf alle Identitäten (Benutzer und Rollen) in einem AWS-Konto zu gestatten oder verweigern. In Schlüsselrichtlinien verwenden Sie das Element `Principal`, um die Identitäten festzulegen, für die die Richtlinieanweisung gilt. Die Syntax für das `Principal`-Element bietet keine Möglichkeit, alle Identitäten in einem AWS-Konto festzulegen. Sie können diesen Effekt jedoch erzielen, indem Sie diesen Bedingungsschlüssel mit einem `Principal` Element kombinieren, das alle AWS Identitäten spezifiziert.

Sie können damit den Zugriff auf jeden KMS-Schlüsselressourcenvorgang steuern, d. h. auf jeden AWS KMS Vorgang, der einen bestimmten KMS-Schlüssel verwendet. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources

(Aktionen und Ressourcen) Sie nach dem Wert von KMS key in der Resources-Spalte für die Operation. Er ist auch gültig für Operationen, die benutzerdefinierte Schlüsselspeicher verwalten.

Die folgende Schlüsselrichtlinienanweisung veranschaulicht beispielsweise die Verwendung des Bedingungsschlüssels kms:CallerAccount. Diese Grundsatzerklärung ist Teil der wichtigsten Richtlinie Von AWS verwalteter Schlüssel für Amazon EBS. Sie kombiniert ein Principal Element, das alle AWS Identitäten spezifiziert, mit dem kms:CallerAccount Bedingungsschlüssel, um effektiv den Zugriff auf alle Identitäten in 111122223333 zu ermöglichen. AWS-Konto Es enthält einen zusätzlichen AWS KMS Bedingungsschlüssel (kms:ViaService), um die Berechtigungen weiter einzuschränken, indem nur Anfragen zugelassen werden, die über Amazon EBS eingehen. Weitere Informationen finden Sie unter km: 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 (veraltet)

Der Bedingungsschlüssel kms:CustomerMasterKeySpec ist veraltet. Verwenden Sie stattdessen den KeySpecBedingungsschlüssel kms:.

Die Bedingungsschlüssel kms:CustomerMasterKeySpec und kms:KeySpec funktionieren auf die gleiche Weise. Nur die Namen unterscheiden sich. Wir empfehlen Ihnen, kms:KeySpec zu

verwenden. AWS KMS Unterstützt jedoch beide Bedingungsschlüssel, um fehlerhafte Änderungen zu vermeiden.

kms: CustomerMasterKeyUsage (veraltet)

Der Bedingungsschlüssel `kms:CustomerMasterKeyUsage` ist veraltet. Verwenden Sie stattdessen den KeyUsageBedingungsschlüssel kms:.

Die Bedingungsschlüssel `kms:CustomerMasterKeyUsage` und `kms:KeyUsage` funktionieren auf die gleiche Weise. Nur die Namen unterscheiden sich. Wir empfehlen Ihnen, `kms:KeyUsage` zu verwenden. AWS KMS Unterstützt jedoch beide Bedingungsschlüssel, um fehlerhafte Änderungen zu vermeiden.

km: DataKeyPairSpec

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:DataKeyPairSpec`	String	Einzelwertig	`GenerateDataKeyPair` `GenerateDataKeyPairWithoutPlaintext`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf die GenerateDataKeyPairWithoutPlaintextOperationen GenerateDataKeyPairund auf der Grundlage des `KeyPairSpec` Parameterwerts in der Anforderung zu steuern. Beispielsweise können Sie Benutzern erlauben, nur bestimmte Typen von Datenschlüsselpaaren zu generieren.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:DataKeyPairSpec`, um es Benutzern zu erlauben, mit dem KMS-Schlüssel nur RSA-Datenschlüsselpaare zu generieren.

```
{
  "Effect": "Allow",
  "Principal": {
```

```
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: KeySpec
- the section called “km: EncryptionAlgorithm”
- the section called “kms:EncryptionContext: Kontextschlüssel”
- the section called “km: EncryptionContextKeys”

km: EncryptionAlgorithm

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:EncryptionAlgorithm`	String	Einzelwertig	`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPai`
				Schlüsselrichtlinien und IAM-Richtlinien

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

			`rWithoutPlaintext`
			
			`GenerateDataKeyWithoutPlaintext`
			
			`ReEncrypt`	

Mit dem Bedingungsschlüssel `kms:EncryptionAlgorithm` können Sie den Zugriff auf kryptografische Operationen basierend auf dem in der Produktion verwendeten Verschlüsselungsalgorithmus kontrollieren. Bei den ReEncryptVorgängen Verschlüsseln, Entschlüsseln und Entschlüsseln steuert es den Zugriff auf der Grundlage des Werts des EncryptionAlgorithmParameters in der Anforderung. Für Operationen, die Datenschlüssel und Datenschlüsselpaare generieren, steuert er den Zugriff basierend auf dem Verschlüsselungsalgorithmus, der zum Verschlüsseln des Datenschlüssels verwendet wird.

Dieser Bedingungsschlüssel hat keine Auswirkung auf Operationen, die außerhalb von ausgeführt werden AWS KMS, wie z. B. die Verschlüsselung mit dem öffentlichen Schlüssel in einem asymmetrischen KMS-Schlüsselpaar außerhalb von. AWS KMS

EncryptionAlgorithm Parameter in einer Anfrage

Damit Benutzer nur einen bestimmten Verschlüsselungsalgorithmus mit einem KMS-Schlüssel verwenden können, verwenden Sie eine Richtlinienanweisung mit dem Effekt `Deny` und dem Bedingungsoperator `StringNotEquals`. Die folgende Schlüsselrichtlinienanweisung verhindert beispielsweise, dass Prinzipale, die die `ExampleRole`-Rolle annehmen können, diesen KMS-Schlüssel in den angegebenen kryptografischen Operationen verwenden, es sei denn, der Verschlüsselungsalgorithmus in der Anforderung ist `RSAES_OAEP_SHA_256` (ein asymmetrischer Verschlüsselungsalgorithmus, der mit RSA-KMS-Schlüsseln verwendet wird).

Im Gegensatz zu einer Richtlinienanweisung, die es einem Benutzer erlaubt, einen bestimmten Verschlüsselungsalgorithmus zu verwenden, hindert eine Richtlinienanweisung mit einem doppelten Negativwert wie diese andere Richtlinien und Erteilungen für diesen KMS-Schlüssel daran, dieser

Rolle die Verwendung anderer Verschlüsselungsalgorithmen zu ermöglichen. Das `Deny` in dieser Schlüsselrichtlinienanweisung hat Vorrang vor allen Schlüsselrichtlinien oder IAM-Richtlinien mit `Allow`-Wirkung, und es hat vor allen Erteilungen für diesen KMS-Schlüssel und seine Prinzipale Vorrang.

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

Der für die Produktion verwendete Verschlüsselungsalgorithmus

Sie können auch den `kms:EncryptionAlgorithm`-Bedingungsschlüssel verwenden, um den Zugriff auf Operationen basierend auf dem in der Produktion verwendeten Verschlüsselungsalgorithmus zu steuern, auch wenn der Algorithmus nicht in der Anforderung angegeben ist. Auf diese Weise können Sie den `SYMMETRIC_DEFAULT`-Algorithmus, der möglicherweise nicht in einer Anforderung angegeben wird, erfordern oder verbieten, da es sich um den Standardwert handelt.

Mit diesem Feature können Sie den `kms:EncryptionAlgorithm`-Bedingungsschlüssel verwenden, um den Zugriff auf die Operationen zu steuern, die Datenschlüssel und Datenschlüsselpaare generieren. Diese Operationen verwenden nur KMS-Schlüssel zur symmetrischen Verschlüsselung und den `SYMMETRIC_DEFAULT`-Algorithmus.

Beispielsweise beschränkt diese IAM-Richtlinie ihre Prinzipale auf symmetrische Verschlüsselung. Sie verweigert den Zugriff auf jeden KMS-Schlüssel im Beispielkonto für kryptografische Operationen, es sei denn, der in der Anforderung angegebene oder in der Produktion verwendete

Verschlüsselungsalgorithmus ist SYMMETRIC_DEFAULT. `GenerateDataKey*`Einschließlich der GenerateDataKeyWithoutPlaintextErweiterungen GenerateDataKey, GenerateDataKeyPair, und GenerateDataKeyPairWithoutPlaintextzu den Berechtigungen. Die Bedingung hat keine Auswirkungen auf diese Vorgänge, da sie immer einen symmetrischen Verschlüsselungsalgorithmus verwenden.

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

Informationen finden Sie auch unter:

- the section called “km: MacAlgorithm”
- km: SigningAlgorithm

kms:EncryptionContext: Kontextschlüssel

AWS KMS Bedingungsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:EncryptionContext:` *`context-key`*	String	Einzelwertig	`CreateGrant`
			`Encrypt`
			`Decrypt`
				Schlüsselrichtlinien und IAM-Richtlinien

AWS KMS Bedingungsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`	

Mit dem `kms:EncryptionContext:`*`context-key`*-Bedingungsschlüssel können Sie den Zugriff auf einen KMS-Schlüssel zu symmetrischen Verschlüsselung anhand des Verschlüsselungskontexts in einer Anforderung für eine kryptografische Produktion steuern. Verwenden Sie diesen Bedingungsschlüssel, um sowohl den Schlüssel als auch den Wert im Verschlüsselungskontext-Paar auszuwerten. Verwenden Sie den EncryptionContextKeys Bedingungsschlüssel kms:, um nur die Schlüssel für den Verschlüsselungskontext auszuwerten oder einen Verschlüsselungskontext unabhängig von Schlüsseln oder Werten zu benötigen.

> **Note**
>
> Bedingungsschlüsselwerte müssen die Zeichenregeln für Schlüsselrichtlinien und IAM-Richtlinien einhalten. Einige Zeichen, die in einem Verschlüsselungskontext gültig sind, sind in Richtlinien nicht gültig. Sie können diesen Bedingungsschlüssel möglicherweise nicht verwenden, um alle gültigen Verschlüsselungskontextwerte auszudrücken. Weitere Informationen zu wichtigen Dokumentenregeln für Schlüsselrichtlinien finden Sie unter

Schlüsselrichtlinienformat. Weitere Informationen zu Regeln für IAM-Richtliniendokumente finden Sie unter Anforderungen für den IAM-Namen im IAM-Benutzerhandbuch.

Sie können mit einem asymmetrische KMS-Schlüssel oder einem HMAC-KMS-Schlüssel keinen Verschlüsselungskontext in einer kryptografischen Produktion angeben. Asymmetrische Algorithmen und MAC-Algorithmen unterstützen keinen Verschlüsselungskontext.

Um den Kontext-Key-Bedingungsschlüssel kmsEncryptionContext:: zu verwenden, ersetzen Sie den Platzhalter für den *Kontextschlüssel durch den Verschlüsselungskontextschlüssel*. Ersetzen Sie den *Kontextwert*-Platzhalter durch den Verschlüsselungskontextwert der Verschlüsselung.

```
"kms:EncryptionContext:context-key": "context-value"
```

Der folgende Bedingungsschlüssel gibt beispielsweise einen Verschlüsselungskontext an, in dem der Schlüssel `AppName` und der Wert `ExampleApp` (`AppName = ExampleApp`) ist.

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Dies ist ein einzelwertiger Bedingungsschlüssel. Der Schlüssel im Bedingungsschlüssel gibt einen bestimmten Verschlüsselungskontext-Schlüssel an (Kontext-Schlüssel). Obwohl Sie mehrere Verschlüsselungskontext-Paare in jede API-Anforderung einschließen können, kann das Verschlüsselungskontext-Paar mit dem angegebenen Kontext-Schlüssel nur einen Wert haben. Der `kms:EncryptionContext:Department`-Bedingungsschlüssel gilt beispielsweise nur für Verschlüsselungskontext-Paare mit einem `Department`-Schlüssel, und jedes gegebene Verschlüsselungskontext-Paar mit dem `Department`-Schlüssel kann nur einen Wert haben.

Verwenden Sie keinen Satz-Operator mit dem `kms:EncryptionContext:`*`context-key`*-Bedingungsschlüssel. Wenn Sie eine Richtlinienanweisung mit einer `Allow`-Aktion, dem `kms:EncryptionContext:`*`context-key`*-Bedingungsschlüssel und dem `ForAllValues`-Satz-Operator erstellen, erlaubt die Bedingung Anforderungen ohne Verschlüsselungskontext und Anforderungen mit Verschlüsselungskontext-Paaren, die nicht in der Richtlinienbedingung angegeben sind.

> **Warning**
>
> Verwenden Sie keinen `ForAnyValue`- oder `ForAllValues`-Satz-Operator mit diesem einzelwertigen Bedingungsschlüssel. Diese Satz-Operatoren können eine Richtlinienbedingung erstellen, die keine der von Ihnen gewünschten Werte erfordert und Werte erlaubt, die Sie verbieten möchten.
> Wenn Sie eine Richtlinie erstellen oder aktualisieren, die einen `ForAllValues` Set-Operator mit dem Kontextschlüssel kms:EncryptionContext: enthält, wird die folgende Fehlermeldung zurückgegeben: AWS KMS `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.`

Um ein bestimmtes Verschlüsselungskontext-Paar zu erfordern, verwenden Sie den `kms:EncryptionContext:`*`context-key`*-Bedingungsschlüssel mit dem `StringEquals`-Operator.

Die folgende Beispiel-Schlüsselrichtlinienanweisung erlaubt es Prinzipalen, die die Rolle übernehmen können, den KMS-Schlüssel in einer `GenerateDataKey`-Anforderung zu verwenden, nur dann, wenn der Verschlüsselungskontext in der Anforderung das `AppName:ExampleApp`-Paar beinhält. Andere Verschlüsselungskontext-Paare sind zulässig.

Für den Schlüsselnamen muss die Groß-/Kleinschreibung nicht berücksichtigt werden. Die Berücksichtigung der Groß-/Kleinschreibung des Wertes wird durch den Bedingungsoperator (z. B. `StringEquals`) festgelegt. Details hierzu finden Sie unter Beachtung der Groß-/Kleinschreibung bei der Verschlüsselungskontextbedingung.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
```

```
    }
  }
}
```

Um ein Verschlüsselungskontextpaar vorzuschreiben und alle anderen Verschlüsselungskontextpaare zu verbieten, verwenden Sie sowohl den kms:EncryptionContext: -Kontextschlüssel als auch in der Richtlinienanweisung. `kms:EncryptionContextKeys` Die folgende Schlüsselrichtlinienanweisung verwendet die `kms:EncryptionContext:AppName`-Bedingung, um das `AppName=ExampleApp`-Verschlüsselungskontext-Paar in der Anforderung zu erfordern. Sie verwendet auch einen `kms:EncryptionContextKeys`-Bedingungsschlüssel mit dem `ForAllValues`-Satz-Operator, um nur den `AppName`-Verschlüsselungskontext-Schlüssel zu erlauben.

Der `ForAllValues`-Satz-Operator beschränkt Verschlüsselungskontext-Schlüssel in der Anforderung an `AppName`. Wenn die `kms:EncryptionContextKeys`-Bedingung mit dem `ForAllValues`-Satz-Operator alleine in einer Richtlinienanweisung verwendet würde, würde dieser Satz-Operator Anforderungen ohne Verschlüsselungskontext erlauben. Wenn die Anforderung jedoch keinen Verschlüsselungskontext hätte, würde die `kms:EncryptionContext:AppName`-Bedingung fehlschlagen. Ausführliche Informationen zu dem `ForAllValues`-Satz-Operator finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch.

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

Sie können diesen Bedingungsschlüssel auch verwenden, um den Zugriff auf einen KMS-Schlüssel für eine bestimmte Produktion zu verweigern. Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet eine `Deny`-Wirkung, um dem Prinzipal die Verwendung des KMS-Schlüssels zu verbieten, wenn der Verschlüsselungskontext in der Anforderung ein `Stage=Restricted`-Verschlüsselungskontext-Paar enthält. Diese Bedingung ermöglicht eine Anforderung mit anderen Verschlüsselungskontext-Paaren, einschließlich Verschlüsselungskontext-Paaren mit dem `Stage`-Schlüssel und andere Werte, wie `Stage=Test`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

Verwenden mehrerer Verschlüsselungskontext-Paare

Sie können mehrere Verschlüsselungskontext-Paare erfordern oder verbieten. Sie können auch eines von mehreren Verschlüsselungskontext-Paaren erfordern. Ausführliche Informationen zur Logik, die zum Interpretieren dieser Bedingungen verwendet wird, finden Sie unter Erstellen einer Bedingung mit mehreren Schlüsseln oder Werten im IAM-Benutzerhandbuch.

> **Note**
>
> In früheren Versionen dieses Themas wurden Richtlinienanweisungen angezeigt, in denen die Operatoren `ForAnyValue` und `ForAllValues` set mit dem Kontext-Schlüssel-Bedingungsschlüssel kms:EncryptionContext: verwendet wurden. Verwenden eines Set-Operators mit einem einzelwertigen Bedingungsschlüssel kann zu Richtlinien führen, die Anforderungen ohne Verschlüsselungskontext und mit nicht-spezifizierten Verschlüsselungskontext-Paaren erlauben.
> Eine Richtlinienbedingung mit dem `Allow`-Effekt, dem `ForAllValues`-Satz-Operator und dem `"kms:EncryptionContext:Department": "IT"`-Bedingungsschlüssel, beschränkt den Verschlüsselungskontext nicht auf das Paar „Department=IT“. Sie erlaubt

Anforderungen ohne Verschlüsselungskontext und Anforderungen mit nicht spezifizierten Verschlüsselungskontext-Paaren, wie `Stage=Restricted`.
Bitte überprüfen Sie Ihre Richtlinien und entfernen Sie den Operator set aus allen Bedingungen mit dem Kontextschlüssel kms:EncryptionContext:. Versuche, eine Richtlinie mit diesem Format zu erstellen oder zu aktualisieren, schlagen mit einer `OverlyPermissiveCondition`-Ausnahme fehl. Um den Fehler zu beheben, löschen Sie den Satz-Operator.

Um mehrere Verschlüsselungskontext-Paare zu erfordern, listen Sie die Paare in derselben Bedingung auf. Die folgende Beispiel-Schlüsselrichtlinienanweisung erfordert zwei Verschlüsselungskontext-Paare, `Department=IT` und `Project=Alpha`. Da die Bedingungen unterschiedliche Schlüssel haben (`kms:EncryptionContext:Department` und `kms:EncryptionContext:Project`), werden sie implizit durch einen AND-Operator verbunden. Andere Verschlüsselungskontext-Paare sind zulässig, aber nicht erforderlich.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    }
  }
}
```

Um ein Verschlüsselungskontext-Paar OR ein anderes Paar zu erfordern, platzieren Sie jeden Bedingungsschlüssel in einer separaten Richtlinienanweisung. Das folgende Beispiel einer Schlüsselrichtlinie erfordert `Department=IT`- oder `Project=Alpha`-Paare oder beides. Andere Verschlüsselungskontext-Paare sind zulässig, aber nicht erforderlich.

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

Um bestimmte Verschlüsselungspaare vorzuschreiben und alle anderen Verschlüsselungskontextpaare auszuschließen, verwenden Sie sowohl kms:EncryptionContext: -Kontextschlüssel als auch `kms:EncryptionContextKeys`in der Richtlinienerklärung. In der folgenden wichtigen Richtlinienanweisung wird die Kontext-Schlüsselbedingung kms:EncryptionContext: verwendet, um einen Verschlüsselungskontext mit beiden UN-Paaren vorzuschreiben. *`Department=IT`* `Project=Alpha` Sie verwendet einen `kms:EncryptionContextKeys`-Bedingungsschlüssel mit dem `ForAllValues`-Satz-Operator, um nur die `Department`- und `Project`-Verschlüsselungskontext-Schlüssel zu erlauben.

Der `ForAllValues`-Satz-Operator beschränkt Verschlüsselungskontext-Schlüssel in der Anforderung an `Department` und `Project`. Wenn er allein in einer Bedingung verwendet würde, würde dieser Set-Operator Anfragen ohne Verschlüsselungskontext zulassen, aber in dieser Konfiguration würde der KMS:EncryptionContext: -Kontextschlüssel in dieser Bedingung fehlschlagen.

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

Sie können auch mehrere Verschlüsselungskontext-Paare verbieten. Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet eine `Deny`-Wirkung, um dem Prinzipal die Verwendung des KMS-Schlüssels zu verbieten, wenn der Verschlüsselungskontext in der Anforderung ein `Stage=Restricted`- oder `Stage=Production`-Paar enthält.

Mehrere Werte (`Restricted` und `Production`) für denselben Schlüssel (`kms:EncryptionContext:Stage`) sind implizit durch ein OR verbunden. Details dazu finden Sie unter Auswertungslogik für Bedingungen mit mehreren Schlüsseln oder Werten im IAM-Benutzerhandbuch.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
         "Restricted",
         "Production"
      ]
    }
```

```
  }
}
```

Beachtung der Groß-/Kleinschreibung bei der Verschlüsselungskontextbedingung

Der Verschlüsselungskontext, der in einem Entschlüsselungsvorgang angegeben wird, muss exakt mit dem Verschlüsselungskontext übereinstimmen, der in dem Verschlüsselungsvorgang angegeben wird (inklusive Groß-/Kleinschreibung). Nur die Reihenfolge, in der die Paare angegeben werden, spielt keine Rolle.

In den Richtlinienbedingungen wird die Groß-/Kleinschreibung für den Bedingungsschlüssel nicht berücksichtigt. Die Berücksichtigung der Groß-/Kleinschreibung des Bedingungswertes wird durch den von Ihnen verwendeten Richtlinienbedingungsoperator (z. B. `StringEquals` oder `StringEqualsIgnoreCase`) festgelegt.

Daher unterscheidet der Bedingungsschlüssel, der aus dem `kms:EncryptionContext:`-Präfix und dem *`context-key`*-Ersatz besteht, nicht zwischen Groß-/Kleinschreibung. Eine Richtlinie, die diese Bedingung verwendet, berücksichtigt die Groß-/Kleinschreibung bei beiden Elemente des Bedingungsschlüssels nicht. Die Berücksichtigung der Groß-/Kleinschreibung des Wertes (d. h. der *`context-value`*-Ersatz) wird durch den Richtlinienbedingungsoperator festgelegt.

Die folgende Richtlinie erlaubt den Vorgang, wenn der Verschlüsselungskontext einen `Appname`-Schlüssel enthält – unabhängig von der Groß-/Kleinschreibung. Die `StringEquals`-Bedingung erfordert, dass `ExampleApp` wie ursprünglich angegeben geschrieben wird.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Appname": "ExampleApp"
    }
  }
}
```

Um einen Verschlüsselungskontextschlüssel mit Berücksichtigung der Groß- und Kleinschreibung zu verlangen, verwenden Sie die EncryptionContextKeys Richtlinienbedingung kms: mit

einem Bedingungsoperator, bei dem Groß- und Kleinschreibung beachtet werden muss, z. B. `StringEquals` Da der Verschlüsselungskontext-Schlüssel in dieser Richtlinienbedingung der Wert der Richtlinienbedingung ist, wird die Groß-/Kleinschreibung durch den Bedingungsoperator berücksichtigt.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Um eine Bewertung sowohl des Schlüssels als auch des Werts des Verschlüsselungskontextes unter Berücksichtigung der Groß- und Kleinschreibung zu verlangen, verwenden Sie die Kontext-Schlüssel-Richtlinienbedingungen *`kms:EncryptionContextKeys`* und kmsEncryptionContext:: zusammen in derselben Richtlinienanweisung. Der Bedingungsoperator (z. B.`StringEquals`) gilt immer für den Wert der Bedingung. Der Verschlüsselungskontext-Schlüssel (z. B.`AppName`) ist der Wert der `kms:EncryptionContextKeys`-Bedingung. Der Wert für den Verschlüsselungskontext (z. B.*`ExampleApp`*) ist der Wert der Kontext-Schlüsselbedingung kms:EncryptionContext:.

In der folgenden Beispiel-Schlüsselrichtlinienanweisung wird, nachdem der `StringEquals`-Operator die Groß-/Kleinschreibung berücksichtigt, sowohl im Schlüssel als auch im Wert des Verschlüsselungskontextes die Groß-/Kleinschreibung berücksichtigt.

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

Verwenden von Variablen in einer Verschlüsselungskontextbedingung

Der Schlüssel und der Wert in einem Verschlüsselungskontextpaar müssen einfache Literalzeichenfolgen sein. Sie dürfen keine Ganzzahlen oder Objekte oder Typen, die nicht vollständig aufgelöst sind, sein. Wenn Sie einen anderen Typ verwenden, z. B. eine Ganzzahl oder eine Fließkommazahl, AWS KMS interpretiert dies als Literalzeichenfolge.

```
"encryptionContext": {
    "department": "10103.0"
}
```

Der Wert im `kms:EncryptionContext:`*`context-key`*-Bedingungsschlüsselpaar kann jedoch eine IAM-Richtlinien-Variable sein. Diese Richtlinien-Variablen werden zur Laufzeit basierend auf Werten in der Anforderung aufgelöst. Beispielsweise wird `aws:CurrentTime` zum Zeitpunkt der Anforderung und `aws:username` zum Anzeigenamen des Aufrufers aufgelöst.

Sie können diese Richtlinien-Variablen verwenden, um eine Richtlinienanweisung mit einer Bedingung zu erstellen, die sehr spezifische Informationen in einem Verschlüsselungskontext erfordert, z. B. den Benutzernamen des Aufrufers. Da sie eine Variable enthält, können Sie dieselbe Richtlinienanweisung für alle Benutzer verwenden, die die Rolle übernehmen können. Sie müssen nicht für jeden Benutzer eine separate Richtlinienanweisung schreiben.

Angenommen, Sie möchten, dass alle Benutzer, die eine Rolle annehmen können, denselben KMS-Schlüssel verwenden, um ihre Daten zu verschlüsseln und zu entschlüsseln. Sie möchten ihnen jedoch nur erlauben, die Daten zu entschlüsseln, die sie verschlüsselt haben. Stellen Sie zunächst fest, dass jede Anfrage einen Verschlüsselungskontext AWS KMS enthalten muss, in dem sich der Schlüssel befindet `user` und der Wert dem AWS Benutzernamen des Aufrufers entspricht, z. B. der folgende.

```
"encryptionContext": {
    "user": "bob"
}
```

Um diese Anforderung zu erzwingen, können Sie dann eine Richtlinienanweisung wie die im folgenden Beispiel verwenden. Diese Richtlinienanweisung erteilt der `TestTeam`-Rolle die Berechtigung zum Verschlüsseln und Entschlüsseln von Daten mit dem KMS-Schlüssel. Die Berechtigung ist jedoch nur gültig, wenn der Verschlüsselungskontext in der Anforderung ein `"user": "`*`<username>`*`"`-Paar enthält. Um den Benutzernamen darzustellen, verwendet die Bedingung die `aws:username`-Richtlinien-Variable.

Wenn die Anforderung ausgewertet wird, ersetzt der Benutzername des Aufrufers die Variable in der Bedingung. So erfordert die Bedingung einen Verschlüsselungskontext von `"user": "bob"` für "bob" und `"user": "alice"` für "alice".

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

Sie können eine IAM-Richtlinien-Variable nur im Wert des Bedingungsschlüssels `kms:EncryptionContext:`*`context-key`* verwenden. Sie können keine Variable im Schlüssel verwenden.

Sie können auch anbieterspezifische Kontextschlüssel in Variablen verwenden. Diese Kontextschlüssel identifizieren eindeutig Benutzer, die sich AWS mithilfe des Web Identity Federation angemeldet haben.

Wie alle Variablen können diese Variablen nur in der `kms:EncryptionContext:`*`context-key`*-Richtlinienbedingung und nicht im eigentlichen Verschlüsselungskontext verwendet werden. Und sie können nur im Wert der Bedingung verwendet werden, nicht im Schlüssel.

Die folgende Schlüsselrichtlinienanweisung ähnelt beispielsweise der vorherigen. Die Bedingung erfordert jedoch einen Verschlüsselungskontext, in dem der Schlüssel `sub` ist und der Wert einen Benutzer eindeutig identifiziert, der in einem Amazon-Cognito-Benutzerpool angemeldet ist. Weitere Informationen zum Identifizieren von Benutzern und Rollen in Amazon Cognito finden Sie unter IAM-Rollen im Amazon-Cognito-Entwicklerhandbuch.

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

Informationen finden Sie auch unter:

- the section called “km: EncryptionContextKeys”
- the section called “km: GrantConstraintType”

km: EncryptionContextKeys

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:EncryptionContextKeys`	Zeichenfolge (Liste)	Mehrwertig	`CreateGrant`
			`Decrypt`
			`Encrypt`
				Schlüsselrichtlinien und IAM-Richtlinien

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`	

Mit dem Bedingungsschlüssel `kms:EncryptionContextKeys` können Sie den Zugriff auf einen KMS-Schlüssel zu symmetrischen Verschlüsselung anhand des Verschlüsselungskontexts in einer Anforderung für eine kryptografische Produktion steuern. Verwenden Sie diesen Bedingungsschlüssel, um nur den Wert in den einzelnen Verschlüsselungskontext-Paaren auszuwerten. Verwenden Sie den Bedingungsschlüssel `kms:EncryptionContext:`*`context-key`*, um sowohl den Schlüssel als auch den Wert im Verschlüsselungskontext auszuwerten.

Sie können mit einem asymmetrische KMS-Schlüssel oder einem HMAC-KMS-Schlüssel keinen Verschlüsselungskontext in einer kryptografischen Produktion angeben. Asymmetrische Algorithmen und MAC-Algorithmen unterstützen keinen Verschlüsselungskontext.

> **Note**
>
> Bedingungsschlüsselwerte, einschließlich eines Verschlüsselungskontextschlüssels, müssen den Zeichen- und Kodierungsregeln für AWS KMS Schlüsselrichtlinien entsprechen. Sie können diesen Bedingungsschlüssel möglicherweise nicht verwenden, um alle gültigen

Verschlüsselungskontextschlüssel auszudrücken. Weitere Informationen zu wichtigen Dokumentenregeln für Schlüsselrichtlinien finden Sie unter Schlüsselrichtlinienformat. Weitere Informationen zu Regeln für IAM-Richtliniendokumente finden Sie unter Anforderungen für den IAM-Namen im IAM-Benutzerhandbuch.

Dies ist ein mehrwertiger Bedingungsschlüssel. Sie können mehrere Verschlüsselungskontext-Paare in jeder API-Anforderung angeben. `kms:EncryptionContextKeys` vergleicht die Verschlüsselungskontext-Schlüssel in der Anforderung mit dem Satz von Verschlüsselungskontext-Schlüsseln in der Richtlinie. Um zu bestimmen, wie diese Sätze verglichen werden, müssen Sie einen `ForAnyValue` oder `ForAllValues`-Satz-Operator in der Richtlinienbedingung angeben. Ausführliche Informationen zu den Satz-Operatoren finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch.

- `ForAnyValue`: Mindestens ein Verschlüsselungskontext-Schlüssel in der Anforderung muss mit einem Verschlüsselungskontext-Schlüssel in der Richtlinienbedingung übereinstimmen. Andere Verschlüsselungskontext-Schlüssel sind zulässig. Wenn die Anforderung keinen Verschlüsselungskontext aufweist, ist die Bedingung nicht erfüllt.
- `ForAllValues`: Mindestens ein Verschlüsselungskontext-Schlüssel in der Anforderung muss mit einem Verschlüsselungskontext-Schlüssel in der Richtlinienbedingung übereinstimmen. Dieser Satz-Operator beschränkt die Verschlüsselungskontext-Schlüssel auf diejenigen in der Richtlinienbedingung. Er erfordert keine Verschlüsselungskontext-Schlüssel, aber er verbietet nicht spezifizierte Verschlüsselungskontext-Schlüssel.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den `kms:EncryptionContextKeys`-Bedingungsschlüssel mit dem `ForAnyValue`-Operator. Diese Richtlinienanweisung erlaubt die Verwendung eines KMS-Schlüssels für die angegebenen Operationen nur, wenn mindestens eines der Verschlüsselungskontext-Paare in der Anforderung den `AppName`-Schlüssel enthält, unabhängig von dessen Wert.

Diese Schlüsselrichtlinienanweisung erlaubt beispielsweise eine `GenerateDataKey`-Anforderung mit zwei Verschlüsselungskontext-Paaren, `AppName=Helper` und `Project=Alpha`, da das erste Verschlüsselungskontext-Paar die Bedingung erfüllt. Eine Anforderung mit nur `Project=Alpha` oder ohne Verschlüsselungskontext würde fehlschlagen.

Da bei der StringEqualsBedingungsoperation Groß- und Kleinschreibung beachtet wird, erfordert diese Richtlinienanweisung die Schreibweise und Groß- und Kleinschreibung des

Verschlüsselungskontextschlüssels. Sie können aber einen Bedingungsoperator verwenden, der die Groß-/Kleinschreibung des Schlüssels ignoriert – z. B. `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Sie können den `kms:EncryptionContextKeys`-Bedingungsschlüssel auch verwenden, um einen (beliebigen) Verschlüsselungskontext in kryptografischen Operationen, die den KMS-Schlüssel verwenden, zu erfordern.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:EncryptionContextKeys` mit dem Null-bedingten Operator, um den Zugriff auf einen KMS-Schlüssel nur zu erlauben, wenn der Verschlüsselungskontext in der API-Anforderung nicht Null ist. Diese Bedingung überprüft nicht die Schlüssel oder Werte des der Verschlüsselungskontexts. Sie überprüft nur, ob der Verschlüsselungskontext vorhanden ist.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
```

```
      "kms:EncryptionContextKeys": false
    }
  }
}
```

Informationen finden Sie auch unter:

- kms:EncryptionContext: Kontextschlüssel
- km: GrantConstraintType

km: ExpirationModel

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ExpirationModel`	String	Einzelwertig	`ImportKeyMaterial`	Schlüsselrichtlinien und IAM-Richtlinien

Der `kms:ExpirationModel` Bedingungsschlüssel steuert den Zugriff auf die ImportKeyMaterialOperation auf der Grundlage des Werts des ExpirationModelParameters in der Anforderung.

`ExpirationModel` ist ein optionaler Parameter, der festlegt, ob das importierte Schlüsselmaterial abgelaufen ist. Gültige Werte sind `KEY_MATERIAL_EXPIRES` und `KEY_MATERIAL_DOES_NOT_EXPIRE`. Der Standardwert ist `KEY_MATERIAL_EXPIRES`.

Das Ablaufdatum und die Uhrzeit werden durch den Wert des ValidToParameters bestimmt. Der `ValidTo`-Parameter ist erforderlich, es sei denn der Wert des `ExpirationModel`-Parameters lautet `KEY_MATERIAL_DOES_NOT_EXPIRE`. Sie können auch den ValidTo Bedingungsschlüssel kms: verwenden, um ein bestimmtes Ablaufdatum als Bedingung für den Zugriff festzulegen.

Das folgende Richtlinienanweisungsbeispiel verwendet den `kms:ExpirationModel`-Bedingungsschlüssel, um Benutzern das Importieren von Schlüsselmaterial in einen KMS-Schlüssel nur dann zu erlauben, wenn die Anforderung den `ExpirationModel`-Parameter enthält und der entsprechende Wert `KEY_MATERIAL_DOES_NOT_EXPIRE` lautet.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

Sie können auch den `kms:ExpirationModel`-Bedingungsschlüssel verwenden, um Benutzern das Importieren von Schlüsselmaterial nur dann zu erlauben, wenn das Schlüsselmaterial abgelaufen ist, ohne ein Ablaufdatum in der Bedingung anzugeben. Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den `kms:ExpirationModel`-Bedingungsschlüssel mit dem Null-bedingten Operator, um Benutzern das Importieren von Schlüsselmaterial nur dann zu erlauben, wenn die Anforderung keinen `ExpirationModel`-Parameter enthält. Der Standardwert für ExpirationModel ist`KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Informationen finden Sie auch unter:

- km: ValidTo
- km: WrappingAlgorithm

- km: WrappingKeySpec

km: GrantConstraintType

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:GrantConstraintType`	String	Einzelwertig	`CreateGrant`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf den CreateGrantVorgang auf der Grundlage der Art der Gewährungsbeschränkung in der Anfrage zu steuern.

Wenn Sie eine Erteilung erstellen, können Sie optional eine Erteilungseinschränkung festlegen, damit die Operationen nur dann Zugriff gewähren, wenn ein bestimmter Verschlüsselungskontext vorhanden ist. Die Erteilungseinschränkung kann einem der folgenden beiden Typen vorliegen: `EncryptionContextEquals` oder `EncryptionContextSubset`. Sie können diesen Bedingungsschlüssel verwenden, um zu überprüfen, ob die Anforderung den einen oder den anderen Typen enthält.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:GrantConstraintType`, um Benutzern nur dann das Erstellen von Erteilungen zu gestatten, wenn die Anforderung die Erteilungseinschränkung `EncryptionContextEquals` enthält. Das Beispiel zeigt eine Richtlinienanweisung in einer Schlüsselrichtlinie.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

Informationen finden Sie auch unter:

- kms:EncryptionContext: Kontextschlüssel
- km: EncryptionContextKeys
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: GranteePrincipal
- km: RetiringPrincipal

km: GrantIsFor AWSResource

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:GrantIsForAWSResource`	Boolesch	Einzelwertig	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				Schlüsselrichtlinien und IAM-Richtlinien

Erlaubt oder verweigert die Erlaubnis für die RevokeGrantOperationen CreateGrantListGrants, oder nur, wenn ein in integrierter AWS Dienst den Vorgang im Namen des Benutzers AWS KMS aufruft. Diese Richtlinienbedingung erlaubt es dem Benutzer nicht, diese Erteilungs-Operationen direkt aufzurufen.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:GrantIsForAWSResource`. Es ermöglicht integrierten AWS Diensten wie Amazon EBS AWS KMS, im Namen des angegebenen Prinzipals Grants für diesen KMS-Schlüssel zu gewähren.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Informationen finden Sie auch unter:

- km: GrantConstraintType
- km: GrantOperations
- km: GranteePrincipal
- km: RetiringPrincipal

km: GrantOperations

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:GrantOperations`	String	Mehrwertig	`CreateGrant`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf den CreateGrantVorgang auf der Grundlage der in der Anfrage enthaltenen Zuschussoperationen zu steuern. Sie können

beispielsweise Benutzer berechtigen, Erteilungen zu erstellen, mit denen die Berechtigung zum Verschlüsseln, aber nicht zum Entschlüsseln gewährt wird. Weitere Informationen zu Erteilungen finden Sie unter Verwenden von Erteilungen.

Dies ist ein mehrwertiger Bedingungsschlüssel. `kms:GrantOperations` vergleicht den Satz von Erteilungs-Operationen in der `CreateGrant`-Anforderung an den Satz der Erteilungs-Operationen in der Richtlinie. Um zu bestimmen, wie diese Sätze verglichen werden, müssen Sie einen `ForAnyValue` oder `ForAllValues`-Satz-Operator in der Richtlinienbedingung angeben. Ausführliche Informationen zu den Satz-Operatoren finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch.

- `ForAnyValue`: Mindestens eine Erteilungs-Operation in der Anforderung muss mit einem der Erteilungs-Operationen in der Richtlinienbedingung übereinstimmen. Andere Erteilungs-Operationen sind zulässig.
- ForAllValues: Jeder Grant-Vorgang in der Anfrage muss mit einem Grant-Vorgang in der Richtlinienbedingung übereinstimmen. Dieser Satz-Operator beschränkt die Erteilungs-Operationen auf die in der Richtlinienbedingung angegebenen Operationen. Er erfordert keine Erteilungs-Operationen, aber er verbietet nicht-spezifizierte Erteilungs-Operationen.

  ForAllValues gibt auch „true“ zurück, wenn die Anfrage keine Zuschussvorgänge enthält, lässt sie aber `CreateGrant` nicht zu. Wenn das `Operations`-Symbol fehlt oder einen Nullwert hat, schlägt die `CreateGrant`-Anforderung fehl.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:GrantOperations`, um nur dann das Erstellen von Erteilungen zu erlauben, wenn die Erteilungs-Operationen `Encrypt`, `ReEncryptTo` oder beides sind. Wenn die Erteilung andere Operationen umfasst, schlägt die `CreateGrant`-Anforderung fehl.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
```

```
        "ReEncryptTo"
      ]
    }
  }
}
```

Wenn Sie den Satz-Operator in der Richtlinienbedingung auf `ForAnyValue` ändern, würde die Richtlinienanweisung erfordern, dass mindestens eine der Erteilungs-Operationen in der Erteilung `Encrypt` oder `ReEncryptTo` ist, aber es würde andere Erteilungs-Operationen wie `Decrypt` oder `ReEncryptFrom` erlauben.

Informationen finden Sie auch unter:

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GranteePrincipal
- km: RetiringPrincipal

km: GranteePrincipal

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:GranteePrincipal`	String	Einzelwertig	`CreateGrant`	IAM- und Schlüsselrichtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf die CreateGrantOperation anhand des GranteePrincipalParameterwerts in der Anforderung zu steuern. Sie können beispielsweise das Erstellen von Erteilungen zur Verwendung eines KMS-Schlüssels nur erlauben, wenn der erteilungsempfangende Prinzipal in der `CreateGrant`-Anforderung dem in der Bedingungsanweisung angegebenen Prinzipal entspricht.

Um den Principal des Empfängers anzugeben, verwenden Sie den Amazon-Ressourcennamen (ARN) eines AWS Prinzipals. Zu den gültigen Prinzipalen gehören AWS-Konten IAM-Benutzer, IAM-

Rollen, Verbundbenutzer und Benutzer mit angenommenen Rollen. Hilfe zur ARN-Syntax für einen Prinzipal finden Sie unter IAM-ARNs im IAM-Benutzerhandbuch.

Die folgende Beispiel-Schlüsselrichtlinienanweisung verwendet den Bedingungsschlüssel `kms:GranteePrincipal`, um nur dann das Erstellen von Erteilungen für einen KMS-Schlüssel zu erlauben, wenn der erteilungsempfangende Prinzipal in der Erteilung die `LimitedAdminRole` ist.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: RetiringPrincipal

km: KeyOrigin

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:KeyOrigin`	String	Einzelwertig	`CreateKey` KMS-Schlüsselressourcen-Operationen	IAM-Richtlinien Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:KeyOrigin` steuert den Zugriff auf Vorgänge anhand des Wertes der `Origin`-Eigenschaft des KMS-Schlüssels, der von der Produktion erstellt oder verwendet wird. Er funktioniert als Ressourcenbedingung oder als Anforderungsbedingung.

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf die CreateKeyOperation auf der Grundlage des Werts des Origin-Parameters in der Anfrage zu steuern. Gültige Werte für `Origin` sind `AWS_KMS`, `AWS_CLOUDHSM` und `EXTERNAL`.

Sie können beispielsweise einen KMS-Schlüssel nur erstellen, wenn das Schlüsselmaterial in AWS KMS (`AWS_KMS`) generiert wird, nur wenn das Schlüsselmaterial in einem AWS CloudHSM Cluster generiert wird, der einem benutzerdefinierten Schlüsselspeicher (`AWS_CLOUDHSM`) zugeordnet ist, oder nur, wenn das Schlüsselmaterial aus einer externen Quelle importiert wird (`EXTERNAL`).

In der folgenden Beispielanweisung für eine Schlüsselrichtlinie wird der `kms:KeyOrigin` Bedingungsschlüssel nur dann verwendet, um einen KMS-Schlüssel zu erstellen, wenn das Schlüsselmaterial AWS KMS erstellt wird.

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

Sie können mit dem Bedingungsschlüssel `kms:KeyOrigin` auch den Zugriff auf Operationen steuern, die einen KMS-Schlüssel verwenden oder verwalten, basierend auf der `Origin`-Eigenschaft des KMS-Schlüssels, der für die Produktion verwendet wird. Bei der Operation muss es sich um eine KMS-Schlüsselressourcen-Operation handeln, das heißt, eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Operation.

Die folgende IAM-Richtlinie erlaubt es beispielsweise Prinzipalen, die angegebenen KMS-Schlüsselressourcen-Operationen auszuführen, jedoch nur mit KMS-Schlüssel in dem Konto, die in einem benutzerdefinierten Schlüsselspeicher erstellt wurde.

```
{
  "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"
```

```
    }
  }
}
```

Informationen finden Sie auch unter:

- km: BypassPolicyLockoutSafetyCheck
- km: KeySpec
- km: KeyUsage

km: KeySpec

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:KeySpec`	String	Einzelwertig	`CreateKey` KMS-Schlüsselressourcen-Operationen	IAM-Richtlinien Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:KeySpec` steuert den Zugriff auf Operationen anhand des Wertes der `KeySpec`-Eigenschaft des KMS-Schlüssels, der von der Produktion erstellt oder verwendet wird.

Sie können diesen Bedingungsschlüssel in einer IAM-Richtlinie verwenden, um den Zugriff auf den CreateKeyVorgang anhand des KeySpecParameterwerts in einer `CreateKey` Anforderung zu steuern. Beispielsweise können Sie es mit dieser Bedingung Benutzern erlauben, nur KMS-Schlüssel für symmetrische Verschlüsselung oder nur HMAC-KMS-Schlüssel zu erstellen.

Die folgende Beispiel-IAM-Richtlinienanweisung verwendet den Bedingungsschlüssel `kms:KeySpec`, um den Prinzipalen nur dann das Erstellen eines asymmetrischen RSA-KMS-Schlüssels zu erlauben. Die Berechtigung ist nur gültig, wenn der `KeySpec` in der Anfrage mit `RSA_` beginnt.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
```

```
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

Sie können mit dem Bedingungsschlüssel `kms:KeySpec` auch den Zugriff auf Operationen steuern, die einen KMS-Schlüssel verwenden oder verwalten, basierend auf der `KeySpec`-Eigenschaft des KMS-Schlüssels, der für die Produktion verwendet wird. Bei der Operation muss es sich um eine KMS-Schlüsselressourcen-Operation handeln, das heißt, eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Produktion.

Die folgende IAM-Richtlinie erlaubt es beispielsweise Prinzipalen, die angegebenen KMS-Schlüsselressourcen-Operationen auszuführen, jedoch nur mit KMS-Schlüsseln zur symmetrischen Verschlüsselung im Konto.

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

Informationen finden Sie auch unter:

- km: BypassPolicyLockoutSafetyCheck
- kms: CustomerMasterKeySpec (veraltet)
- km: DataKeyPairSpec

- km: KeyOrigin
- km: KeyUsage

km: KeyUsage

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:KeyUsage`	String	Einzelwertig	`CreateKey` KMS-Schlüsselressourcen-Operationen	IAM-Richtlinien Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:KeyUsage` steuert den Zugriff auf Operationen anhand des Wertes der `KeyUsage`-Eigenschaft des KMS-Schlüssels, der von der Produktion erstellt oder verwendet wird.

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf die CreateKeyOperation anhand des KeyUsageParameterwerts in der Anforderung zu steuern. Gültige Werte für `KeyUsage` sind `ENCRYPT_DECRYPT`, `SIGN_VERIFY` und `GENERATE_VERIFY_MAC`.

Beispielsweise können Sie es erlauben, einen KMS-Schlüssel nur dann zu erstellen, wenn `KeyUsage` auf `ENCRYPT_DECRYPT` eingestellt ist, oder einem Benutzer die Berechtigung verweigert, wenn `KeyUsage` auf `SIGN_VERIFY` eingestellt ist.

Die folgende Beispiel-IAM-Richtlinienanweisung verwendet den Bedingungsschlüssel `kms:KeyUsage`, damit nur dann ein KMS-Schlüssel erstellt werden kann, wenn `KeyUsage` auf `ENCRYPT_DECRYPT` eingestellt ist.

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

```
  }
}
```

Sie können mit dem Bedingungsschlüssel `kms:KeyUsage` auch den Zugriff auf Operationen steuern, die einen KMS-Schlüssel verwenden oder verwalten, basierend auf der `KeyUsage`-Eigenschaft des KMS-Schlüssels, der für die Produktion verwendet wird. Bei der Operation muss es sich um eine KMS-Schlüsselressourcen-Operation handeln, das heißt, eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Operation.

Die folgende IAM-Richtlinie erlaubt es beispielsweise Prinzipalen, die angegebenen KMS-Schlüsselressourcen-Operationen auszuführen, jedoch nur mit KMS-Schlüsseln im Konto, die für Signatur und Verifizierung verwendet werden.

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

Informationen finden Sie auch unter:

- km: BypassPolicyLockoutSafetyCheck
- kms: CustomerMasterKeyUsage (veraltet)
- km: KeyOrigin
- km: KeySpec

km: MacAlgorithm

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:MacAlgorithm`	String	Einzelwertig	`GenerateMac` `VerifyMac`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können den `kms:MacAlgorithm` Bedingungsschlüssel verwenden, um den Zugriff auf die VerifyMacOperationen GenerateMacund basierend auf dem Wert des `MacAlgorithm` Parameters in der Anforderung zu steuern.

Die folgende Beispiel-Schlüsselrichtlinie ermöglicht Benutzern, die zur Verwendung des HMAC-KMS-Schlüssels zum Generieren und Verifizieren von HMAC-Tags nur dann, wenn der MAC-Algorithmus in der Anforderung `HMAC_SHA_384` oder `HMAC_SHA_512` ist, die Rolle `testers` annehmen können. Diese Richtlinie verwendet zwei separate Richtlinienaussagen mit jeweils einer eigenen Bedingung. Wenn Sie mehr als einen MAC-Algorithmus in einer einzigen Bedingungsanweisung angeben, erfordert die Bedingung beide Algorithmen anstelle des einen oder anderen.

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

Informationen finden Sie auch unter:

- the section called “km: EncryptionAlgorithm”
- km: SigningAlgorithm

km: MessageType

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:MessageType`	String	Einzelwertig	`Sign` `Verify`	Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:MessageType` steuert den Zugriff auf die Operationen Sign und Verify basierend auf dem Wert des `MessageType`-Parameters in der Anforderung. Gültige Werte für `MessageType` sind `RAW` und `DIGEST`.

Die folgende Schlüsselrichtlinienanweisung verwendet beispielsweise den Bedingungsschlüssel `kms:MessageType`, um zu erlauben, einen asymmetrischen KMS-Schlüssel zum Signieren einer Nachricht zu verwenden, jedoch keinen Nachrichten-Digest.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:MessageType": "RAW"
    }
  }
}
```

Informationen finden Sie auch unter:

- the section called “km: SigningAlgorithm”

km: MultiRegion

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:MultiRegion`	Boolesch	Einzelwertig	`CreateKey` KMS-Schlüsselressourcen-Operationen	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um Operationen entweder nur für einzelregionale Schlüssel oder nur für multiregionale Schlüssel zu erlauben. Der `kms:MultiRegion` Bedingungsschlüssel steuert den Zugriff auf AWS KMS Operationen mit KMS-Schlüsseln und auf den CreateKeyVorgang, der auf dem Wert der `MultiRegion` Eigenschaft des KMS-Schlüssels basiert.

Gültige Werte sind `true` (multiregionaler Schlüssel) oder `false` (einzelregionaler Schlüssel). Alle KMS-Schlüssel verfügen über eine `MultiRegion`-Eigenschaft.

Das folgende Beispiel einer IAM-Richtlinienanweisung verwendet den `kms:MultiRegion`-Bedingungsschlüssel, um den Prinzipalen nur dann das Erstellen von einzelregionalen Schlüsseln zu gestatten.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}
```

km: MultiRegionKeyType

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:MultiRegionKeyType`	String	Einzelwertig	`CreateKey` KMS-Schlüsselressourcen-Operationen	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um Operationen entweder nur für multiregionale Primärschlüssel oder nur für multiregionale Replikatschlüssel zu erlauben. Der `kms:MultiRegionKeyType` Bedingungsschlüssel steuert den Zugriff auf AWS KMS Operationen mit KMS-Schlüsseln und den CreateKeyVorgang, der auf der `MultiRegionKeyType` Eigenschaft des KMS-Schlüssels basiert. Die gültigen Werte sind `PRIMARY` und `REPLICA`. Nur multiregionale Schlüssel verfügen über eine `MultiRegionKeyType`-Eigenschaft.

In der Regel verwenden Sie den `kms:MultiRegionKeyType`-Bedingungsschlüssel in einer IAM-Richtlinie, um den Zugriff auf mehrere KMS-Schlüssel zu kontrollieren. Da ein bestimmter

multiregionaler Schlüssel jedoch zu Primär- oder Replikat wechseln kann, sollten Sie diese Bedingung in einer Schlüsselrichtlinie verwenden, um eine Produktion nur dann zu erlauben, wenn der bestimmte multiregionale Schlüssel ein Primär- oder Replikatschlüssel ist.

In diesem Beispiel verwendet die IAM-Richtlinienanweisung den Bedingungsschlüssel `kms:MultiRegionKeyType`, um Prinzipalen das Planen und Abbrechen einer Schlüssellöschung nur für multiregionale Replikatschlüssel im angegebenen AWS-Konto zu erlauben.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}
```

Um den Zugriff auf alle multiregionale Schlüssel zu erlauben oder zu verweigern, können Sie beide Werte oder einen Nullwert mit `kms:MultiRegionKeyType` verwenden. Zu diesem Zweck wird jedoch der MultiRegion Bedingungsschlüssel kms: empfohlen.

km: PrimaryRegion

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:PrimaryRegion`	Zeichenfolge (Liste)	Einzelwertig	`UpdatePrimaryRegion`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um die Zielregionen in einem UpdatePrimaryRegionVorgang einzuschränken. Diese AWS-Regionen können Ihre Primärschlüssel für mehrere Regionen hosten.

Der `kms:PrimaryRegion` Bedingungsschlüssel steuert den Zugriff auf die UpdatePrimaryRegionOperation auf der Grundlage des `PrimaryRegion` Parameterwerts. Der `PrimaryRegion` Parameter gibt den AWS-Region Replikatschlüssel für mehrere Regionen an, der zum Primärschlüssel heraufgestuft wird. Der Wert der Bedingung besteht aus einem oder mehreren AWS-Region Namen, z. B. `us-east-1` oder`ap-southeast-2`, oder Regionsnamenmustern, wie `eu-*`

Die folgende Schlüsselrichtlinienanweisung verwendet beispielsweise den Bedingungsschlüssel `kms:PrimaryRegion`, um es Prinzipalen zu erlauben, die primäre Region eines multiregionalen Schlüssels auf eine der vier angegebenen Regionen zu aktualisieren.

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

km: ReEncryptOnSameKey

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ReEncryptOnSameKey`	Boolesch	Einzelwertig	`ReEncrypt`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf den ReEncryptVorgang zu steuern, je nachdem, ob in der Anforderung ein KMS-Zielschlüssel angegeben ist, der derselbe ist, der für die ursprüngliche Verschlüsselung verwendet wurde.

Die folgende Richtlinienanweisung verwendet beispielsweise den Bedingungsschlüssel `kms:ReEncryptOnSameKey`, um zu erlauben, nur dann eine erneute Verschlüsselung vorzunehmen, wenn der verwendete Ziel-KMS-Schlüssel mit dem der ursprünglichen Verschlüsselung übereinstimmt.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}
```

km: RequestAlias

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:RequestAlias`	Zeichenfolge (Liste)	Einzelwertig	Kryptografische Operationen DescribeKey GetPublicKey	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um eine Produktion nur dann zu erlauben, wenn die Anforderung einen bestimmten Alias zum Identifizieren des KMS-Schlüssels verwendet. Der Bedingungsschlüssel `kms:RequestAlias` steuert den Zugriff auf einen KMS-Schlüssel, der in einer

kryptografischen Produktion verwendet wird, `GetPublicKey`, oder `DescribeKey`, basierend auf dem Alias, der diesen KMS-Schlüssel in der Anforderung identifiziert. (Diese Richtlinienbedingung hat keine Auswirkung auf den GenerateRandomVorgang, da der Vorgang keinen KMS-Schlüssel oder Alias verwendet.)

Diese Bedingung unterstützt die attributebasierte Zugriffskontrolle (ABAC) AWS KMS, mit der Sie den Zugriff auf KMS-Schlüssel anhand der Tags und Aliase eines KMS-Schlüssels steuern können. Sie können Tags und Aliase verwenden, um den Zugriff auf einen KMS-Schlüssel zu erlauben oder zu verweigern, ohne Richtlinien oder Erteilungen zu ändern. Details hierzu finden Sie unter ABAC für AWS KMS.

Um den Alias in dieser Richtlinienbedingung anzugeben, verwenden Sie einen Aliasnamen, wie `alias/project-alpha`, oder ein Alias-Namensmuster, wie `alias/*test*`. Sie können keinen Alias-ARN im Wert dieses Bedingungsschlüssels angeben.

Um diese Bedingung zu erfüllen, muss der Wert des `KeyId`-Parameters in der Anforderung ein übereinstimmender Aliasname oder Alias-ARN sein. Wenn die Anforderung einen anderen Schlüsselbezeichner verwendet, erfüllt er die Bedingung nicht, selbst wenn er denselben KMS-Schlüssel identifiziert.

Die folgende wichtige Richtlinienanweisung ermöglicht es dem Principal beispielsweise, den GenerateDataKeyVorgang mit dem KMS-Schlüssel aufzurufen. Dies ist jedoch nur zulässig, wenn der Wert des `KeyId`-Parameters in der Anforderung `alias/finance-key` ist oder ein Alias-ARN mit diesem Aliasnamen, z. B. `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"
    }
  }
}
```

Sie können diesen Bedingungsschlüssel nicht verwenden, um den Zugriff auf Aliasoperationen wie CreateAliasoder zu steuern DeleteAlias. Weitere Hinweise zum Steuern des Zugriffs auf Alias-Operationen finden Sie unter Steuern des Zugriffs auf Aliasse.

km: ResourceAliases

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ResourceAliases`	Zeichenfolge (Liste)	Mehrwertig	KMS-Schlüsselressourcen-Operationen	Nur IAM-Richtlinien

Verwenden Sie diesen Bedingungsschlüssel, um den Zugriff auf einen KMS-Schlüssel basierend auf den Aliasen, die dem KMS-Schlüssel zugeordnet sind. Bei der Operation muss es sich um eine KMS-Schlüsselressourcen-Operation handeln, das heißt, eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Produktion.

Diese Bedingung unterstützt attributbasierte Zugriffssteuerung (ABAC) in AWS KMS. Mit ABAC können Sie den Zugriff auf KMS-Schlüssel anhand der Tags steuern, die einem KMS-Schlüssel zugewiesen sind, und den Aliasen, die einem KMS-Schlüssel zugeordnet sind. Sie können Tags und Aliasse verwenden, um den Zugriff auf einen KMS-Schlüssel zu erlauben oder zu verweigern, ohne Richtlinien oder Erteilungen zu ändern. Details hierzu finden Sie unter ABAC für AWS KMS.

Ein Alias muss in einer AWS-Konto UND-Region eindeutig sein. Mit dieser Bedingung können Sie jedoch den Zugriff auf mehrere KMS-Schlüssel in derselben Region (mithilfe des `StringLike` Vergleichsoperators) oder auf mehrere KMS-Schlüssel in verschiedenen AWS-Regionen Konten steuern.

> **Note**
>
> Die kms: ResourceAliases -Bedingung ist nur wirksam, wenn der KMS-Schlüssel den Aliasnamen pro KMS-Schlüsselkontingent entspricht. Wenn ein KMS-Schlüssel dieses Kontingent überschreitet, wird auch Prinzipalen, die berechtigt sind, den KMS-Schlüssel zu

nutzen, durch die Bedingung `kms:ResourceAliases` der Zugriff auf den KMS-Schlüssel verweigert.

Um den Alias in dieser Richtlinienbedingung anzugeben, verwenden Sie einen Aliasnamen, wie `alias/project-alpha`, oder ein Alias-Namensmuster, wie `alias/*test*`. Sie können keinen Alias-ARN im Wert dieses Bedingungsschlüssels angeben. Um die Bedingung zu erfüllen, muss der in der Produktion verwendete KMS-Schlüssel über den angegebenen Alias verfügen. Es spielt keine Rolle, ob oder wie der KMS-Schlüssel in der Anforderung für die Produktion identifiziert wird.

Dies ist ein mehrwertiger Bedingungsschlüssel, der den Satz von Aliasen, die einem KMS-Schlüssel zugeordnet sind, mit dem Satz von Aliasen in der Richtlinie vergleicht. Um zu bestimmen, wie diese Sätze verglichen werden, müssen Sie einen `ForAnyValue` oder `ForAllValues`-Satz-Operator in der Richtlinienbedingung angeben. Ausführliche Informationen zu den Satz-Operatoren finden Sie unter Verwenden mehrerer Schlüssel und Werte im IAM-Benutzerhandbuch.

- ForAnyValue: Mindestens ein dem KMS-Schlüssel zugeordneter Alias muss mit einem Alias in der Richtlinienbedingung übereinstimmen. Andere Aliase sind zulässig. Wenn der KMS-Schlüssel keine Aliase aufweist, ist die Bedingung nicht erfüllt.
- ForAllValues: Jeder Alias, der dem KMS-Schlüssel zugeordnet ist, muss mit einem Alias in der Richtlinie übereinstimmen. Dieser Satz-Operator beschränkt die Aliase, die dem KMS-Schlüssel zugeordnet sind, auf diejenigen in der Richtlinienbedingung. Er erfordert keine Aliase, aber er verbietet nicht-spezifizierte Aliase.

Die folgende IAM-Richtlinienanweisung ermöglicht es dem Principal beispielsweise, den GenerateDataKeyVorgang für jeden KMS-Schlüssel in der angegebenen Datei aufzurufen AWS-Konto , der dem `finance-key` Alias zugeordnet ist. (Die Schlüsselrichtlinien der betroffenen KMS-Schlüssel müssen es auch dem Konto des Prinzipals erlauben, sie für diese Produktion zu verwenden.) Um anzuzeigen, dass die Bedingung erfüllt ist, wenn einer der vielen Aliase, die dem KMS-Schlüssel zugeordnet werden könnten, `alias/finance-key` ist, verwendet die Bedingung den `ForAnyValue`-Satz-Operator.

Da die `kms:ResourceAliases`-Bedingung auf der Ressource und nicht auf der Anforderung basiert, ist ein Aufruf an `GenerateDataKey` für jeden KMS-Schlüssel erfolgreich, der dem `finance-key`-Alias zugeordnet ist, auch wenn die Anforderung eine Schlüssel-ID oder einen Schüssel-ARN verwendet, um den KMS-Schlüssel zu identifizieren.

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

Die folgende Beispiel-IAM-Richtlinienanweisung erlaubt es dem Prinzipal, KMS-Schlüssel zu aktivieren und zu deaktivieren, aber nur, wenn alle Aliase den KMS-Schlüssel "`Test`" enthalten. Diese Richtlinienanweisung verwendet zwei Bedingungen. Die Bedingung mit dem `ForAllValues`-Satz-Operator erfordert, dass alle Aliase, die dem KMS-Schlüssel zugeordnet sind, "Test" enthalten. Die Bedingung mit dem `ForAnyValue`-Satz-Operator erfordert, dass der KMS-Schlüssel mindestens einen Alias mit "Test" enthält. Ohne die `ForAnyValue`-Bedingung, hätte diese Richtlinienanweisung es dem Prinzipal erlaubt, KMS-Schlüssel zu verwenden, die keine Aliase hatten.

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

```
  }
}
```

km: ReplicaRegion

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ReplicaRegion`	Zeichenfolge (Liste)	Einzelwertig	`ReplicateKey`	Schlüsselrichtlinien und IAM-Richtlinien

Mit diesem Bedingungsschlüssel können Sie einschränken, AWS-Regionen in welchem Umfang ein Prinzipal einen Schlüssel für mehrere Regionen replizieren kann. Der `kms:ReplicaRegion` Bedingungsschlüssel steuert den Zugriff auf den ReplicateKeyVorgang auf der Grundlage des Werts des ReplicaRegionParameters in der Anforderung. Dieser Parameter gibt die AWS-Region für den neuen Replikatschlüssel an.

Der Wert der Bedingung besteht aus einem oder mehreren AWS-Region Namen, z. B. `us-east-1` oder`ap-southeast-2`, oder aus Namensmustern wie`eu-*`. Eine Liste der AWS KMS unterstützten Namen finden Sie unter AWS Key Management Service Endpunkte und Kontingente in der Allgemeine AWS-Referenz. AWS-Regionen

In der folgenden wichtigen Richtlinienanweisung wird beispielsweise der `kms:ReplicaRegion` Bedingungsschlüssel verwendet, damit Prinzipale den ReplicateKeyVorgang nur aufrufen können, wenn der Wert des `ReplicaRegion` Parameters einer der angegebenen Regionen entspricht.

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

Dieser Bedingungsschlüssel steuert nur den Zugriff auf den ReplicateKeyVorgang. Um den Zugriff auf den UpdatePrimaryRegionVorgang zu steuern, verwenden Sie den PrimaryRegion Bedingungsschlüssel kms:.

km: RetiringPrincipal

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:RetiringPrincipal`	Zeichenfolge (Liste)	Einzelwertig	`CreateGrant`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf die CreateGrantOperation anhand des RetiringPrincipalParameterwerts in der Anforderung zu steuern. Sie können beispielsweise das Erstellen von Erteilungen zur Verwendung eines KMS-Schlüssels nur erlauben, wenn der `RetiringPrincipal` in der `CreateGrant`-Anforderung dem in der Bedingungsanweisung angegebenen `RetiringPrincipal` entspricht.

Um den ausscheidenden Prinzipal anzugeben, verwenden Sie den Amazon-Ressourcennamen (ARN) eines AWS Prinzipals. Zu den gültigen Prinzipalen gehören AWS-Konten IAM-Benutzer, IAM-Rollen, Verbundbenutzer und Benutzer mit angenommenen Rollen. Hilfe zur ARN-Syntax für einen Prinzipal finden Sie unter IAM-ARNs im IAM-Benutzerhandbuch.

Das folgende Beispiel für eine wichtige Richtlinienanweisung ermöglicht es einem Benutzer, Berechtigungen für den KMS-Schlüssel zu erstellen. Der `kms:RetiringPrincipal` Bedingungsschlüssel beschränkt die Berechtigung auf `CreateGrant` Anfragen, bei denen der ausscheidende Schulleiter im Zuschuss der ist. `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"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: GranteePrincipal

km: RotationPeriodInDays

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:RotationPeriodInDays`	Numerischer Wert	Einzelwertig	`EnableKeyRotation`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um die Werte einzuschränken, die Prinzipale im `RotationPeriodInDays` Parameter einer EnableKeyRotationAnfrage angeben können.

Der `RotationPeriodInDays` gibt die Anzahl der Tage zwischen den einzelnen automatischen Schlüsselrotationsdaten an. AWS KMS ermöglicht es Ihnen, einen Rotationszeitraum zwischen 90 und 2560 Tagen anzugeben, aber Sie können die `kms:RotationPeriodInDays` Bedingungstaste verwenden, um den Rotationszeitraum weiter einzuschränken, indem Sie beispielsweise eine Mindestrotationsperiode innerhalb des gültigen Bereichs erzwingen.

In der folgenden wichtigen Richtlinienanweisung wird beispielsweise der `kms:RotationPeriodInDays` Bedingungsschlüssel verwendet, um zu verhindern, dass Prinzipale die Schlüsselrotation aktivieren, wenn der Rotationszeitraum 180 Tage oder weniger beträgt.

```
{
  "Effect": "Deny",
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:RotationPeriodInDays" : "180"
      }
  }
}
```

km: ScheduleKeyDeletionPendingWindowInDays

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ScheduleKeyDeletionPendingWindowInDays`	Numerischer Wert	Einzelwertig	`ScheduleKeyDeletion`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können diesen Bedingungsschlüssel verwenden, um die Werte einzuschränken, die Prinzipale im `PendingWindowInDays` Parameter einer ScheduleKeyDeletionAnfrage angeben können.

Der `PendingWindowInDays` gibt die Anzahl der Tage an, nach denen ein Schlüssel gelöscht AWS KMS wird. AWS KMS ermöglicht es Ihnen, eine Wartezeit zwischen 7 und 30 Tagen anzugeben, aber Sie können den `kms:ScheduleKeyDeletionPendingWindowInDays` Bedingungsschlüssel verwenden, um die Wartezeit weiter einzuschränken, indem Sie beispielsweise eine Mindestwartezeit innerhalb des gültigen Bereichs erzwingen.

Die folgende Schlüsselrichtlinienanweisung verwendet beispielsweise den `kms:ScheduleKeyDeletionPendingWindowInDays`-Bedingungsschlüssel, um zu verhindern,

dass Prinzipale das Löschen von Schlüsseln planen, wenn die Wartezeit weniger als oder gleich 21 Tage beträgt.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
      }
  }
}
```

km: SigningAlgorithm

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:SigningAlgorithm`	String	Einzelwertig	`Sign` `Verify`	Schlüsselrichtlinien und IAM-Richtlinien

Sie können den `kms:SigningAlgorithm` Bedingungsschlüssel verwenden, um den Zugriff auf die Vorgänge Signieren und Verifizieren auf der Grundlage des SigningAlgorithmParameterwerts in der Anforderung zu steuern. Dieser Bedingungsschlüssel hat keine Auswirkung auf Operationen, die außerhalb von ausgeführt werden AWS KMS, wie z. B. die Überprüfung von Signaturen mit dem öffentlichen Schlüssel in einem asymmetrischen KMS-Schlüsselpaar außerhalb von. AWS KMS

Die folgende Beispiel-Schlüsselrichtlinie erlaubt es Benutzern, die die `testers`-Rolle annehmen können, den KMS-Schlüssel zum Signieren von Nachrichten nur dann zu verwenden, wenn der für die Anforderung verwendete Signaturalgorithmus ein RSASSA_PSS-Algorithmus ist, z. B `RSASSA_PSS_SHA512`.

```
{
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: EncryptionAlgorithm
- the section called “km: MacAlgorithm”
- the section called “km: MessageType”

km: ValidTo

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ValidTo`	Zeitstempel	Einzelwertig	`ImportKeyMaterial`	Schlüsselrichtlinien und IAM-Richtlinien

Der `kms:ValidTo` Bedingungsschlüssel steuert den Zugriff auf die ImportKeyMaterialOperation auf der Grundlage des Werts des ValidToParameters in der Anforderung, der bestimmt, wann das importierte Schlüsselmaterial abläuft. Der Wert wird im Unix-Zeitformat angegeben.

Standardmäßig ist der `ValidTo`-Parameter in einer `ImportKeyMaterial`-Anforderung erforderlich. Wenn der Wert des ExpirationModelParameters jedoch ist`KEY_MATERIAL_DOES_NOT_EXPIRE`, ist der `ValidTo` Parameter ungültig. Sie können auch den ExpirationModel Bedingungsschlüssel kms: verwenden, um den `ExpirationModel` Parameter oder einen bestimmten Parameterwert anzufordern.

Die folgende Beispiel-Richtlinienanweisung erlaubt es einem Benutzer, Schlüsselmaterial in einen KMS-Schlüssel zu importieren. Der `kms:ValidTo`-Bedingungsschlüssel beschränkt die Berechtigung für `ImportKeyMaterial`-Anforderungen, bei denen der `ValidTo`-Wert kleiner oder gleich `1546257599.0` ist (31. Dezember 2018 23:59:59).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: ExpirationModel
- km: WrappingAlgorithm
- km: WrappingKeySpec

km: ViaService

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:ViaService`	String	Einzelwertig	KMS-Schlüsselressourcen-Operationen	Schlüsselrichtlinien und IAM-Richtlinien

Der `kms:ViaService` Bedingungsschlüssel beschränkt die Verwendung eines KMS-Schlüssels auf Anfragen von bestimmten AWS Diensten. In jedem `kms:ViaService`-Bedingungsschlüssel können Sie einen oder mehrere Services angeben. Bei der Operation muss es sich um eine KMS-

Schlüsselressourcen-Operation handeln, das heißt, eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen Sie in der Tabelle Actions and Resources (Aktionen und Ressourcen) Sie nach dem Wert von `KMS key` in der `Resources`-Spalte für die Operation.

Die folgende Schlüsselrichtlinienanweisung verwendet beispielsweise den Bedingungsschlüssel `kms:ViaService`, um die Verwendung eines kundenverwalteten KMS-Schlüssels nur für die angegebenen Aktionen zu erlauben, wenn die Anforderung im Auftrag von Amazon EC2 oder Amazon RDS in der Region USA West (Oregon) im Namen von `ExampleRole` ausgegeben wurde.

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

Außerdem können Sie mit dem Bedingungsschlüssel `kms:ViaService` die Berechtigung zur Verwendung eines KMS-Schlüssels verweigern, wenn die Anforderung von bestimmten Services stammt. Die folgende Richtlinienanweisung einer Schlüsselrichtlinie verwendet beispielsweise einen `kms:ViaService`-Bedingungsschlüssel, um zu verhindern, dass ein kundenverwalteter KMS-Schlüssel für `Encrypt`-Operationen verwendet wird, wenn die Anforderung im Namen von `ExampleRole` von AWS Lambda ausgegeben wird.

```
{
  "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

Wenn Sie den Bedingungsschlüssel `kms:ViaService` verwenden, gibt der Service die Anforderung im Namen eines Prinzipals im AWS-Konto aus. Diese Prinzipale müssen über die folgenden Berechtigungen verfügen:

- Berechtigung zur Verwendung des KMS-Schlüssels Der Prinzipal muss dem integrierten Service diese Berechtigungen erteilen, sodass der Service den kundenverwalteten Schlüssel im Auftrag des Prinzipals verwenden kann. Weitere Informationen finden Sie unter Verwendung von AWS KMS durch AWS-Service.
- Berechtigung zur Verwendung des integrierten Service. Einzelheiten dazu, wie Sie Benutzern Zugriff auf einen AWS Dienst gewähren, der in den integrierten Dienst integriert ist AWS KMS, finden Sie in der Dokumentation zum integrierten Dienst.

Alle Von AWS verwaltete Schlüssel verwenden einen `kms:ViaService`-Bedingungsschlüssel in ihrem Schlüsselrichtliniendokument. Diese Bedingung erlaubt es dem KMS-Schlüssel, nur für Anforderungen verwendet zu werden, die von dem Service stammen, der den KMS-Schlüssel erstellt hat. Um die wichtigsten Richtlinien für einen zu sehen Von AWS verwalteter Schlüssel, verwenden Sie den GetKeyPolicyVorgang.

Der Bedingungsschlüssel `kms:ViaService` ist in IAM- und Schlüsselrichtlinienanweisungen gültig. Die von Ihnen angegebenen Services müssen inin AWS KMS integriert sein und den Bedingungsschlüssel `kms:ViaService` unterstützen.

Services, die den Bedingungsschlüssel **`kms:ViaService`** unterstützen

In der folgenden Tabelle sind AWS Dienste aufgeführt, die in den `kms:ViaService` Bedingungsschlüssel integriert sind AWS KMS und dessen Verwendung in vom Kunden verwalteten Schlüsseln unterstützen. Die Dienste in dieser Tabelle sind möglicherweise nicht in allen Regionen verfügbar. Verwenden Sie das `.amazonaws.com` Suffix des AWS KMS ViaService Namens in allen AWS Partitionen.

> **Note**
>
> Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in dieser Tabelle anzuzeigen.

Service-Name	AWS KMS ViaService Name
AWS App Runner	`apprunner.` *`AWS_region`* `.amazonaws.com`
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`

Service-Name	AWS KMS ViaService Name
AWS Backup Gateway	backup-gateway. *AWS_region* .amazonaws.com
Amazon Chime SDK	chimevoiceconnector. *AWS_region* .amazonaws.com
AWS CodeArtifact	codeartifact. *AWS_region* .amazonaws.com
CodeGuru Amazon-Rezensent	codeguru-reviewer. *AWS_region* .amazonaws.com
Amazon Comprehend	comprehend. *AWS_region* .amazonaws.com
Amazon Connect	connect.*AWS_region* .amazonaws.com
Amazon Connect Customer Profiles	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 Directory Service	directoryservice. *AWS_region* .amazonaws.com
Amazon-DynamoDB	dynamodb. *AWS_region* .amazonaws.com
Amazon DocumentDB	docdb-elastic. *AWS_region* .amazonaws.com
Amazon EC2 Systems Manager (SSM)	ssm.*AWS_region* .amazonaws.com
Amazon Elastic Block Store (Amazon EBS)	ec2.*AWS_region* .amazonaws.com (Nur EBS)

Service-Name	AWS KMS ViaService Name
Amazon Elastic Container Registry (Amazon ECR)	`ecr.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic File System (Amazon EFS)	`elasticfilesystem.`*`AWS_region`*` .amazonaws.com`
Amazon ElastiCache	Nehmen Sie beide ViaService Namen in den Wert des Bedingungsschlüssels auf: • `elasticache.`*`AWS_region`*` .amazonaws.com` • `dax.`*`AWS_region`*` .amazonaws.com`
AWS Elemental MediaTailor	`mediatailor.`*`AWS_region`*` .amazonaws.com`
AWS Auflösung der Entität	`entityresolution.`*`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`

Service-Name	AWS KMS ViaService Name
AWS IoT SiteWise	`iotsitewise.` *`AWS_region`* `.amazonaws.com`
Amazon Kendra	`kendra.`*`AWS_region`* `.amazonaws.com`
Amazon Keyspaces (für Apache Cassandra)	`cassandra.` *`AWS_region`* `.amazonaws.com`
Amazon Kinesis	`kinesis.`*`AWS_region`* `.amazonaws.com`
Amazon Data Firehose	`firehose.` *`AWS_region`* `.amazonaws.com`
Amazon Kinesis Video Streams	`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`
Amazon Location Service	`geo.`*`AWS_region`* `.amazonaws.com`
Amazon Lookout für Equipment	`lookoutequipment.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout für Metrics	`lookoutmetrics.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout für Vision	`lookoutvision.` *`AWS_region`* `.amazonaws.com`
Amazon Macie	`macie.`*`AWS_region`* `.amazonaws.com`
AWS Mainframe Modernization	`m2.`*`AWS_region`* `.amazonaws.com`

Service-Name	AWS KMS ViaService Name
Amazon Managed Blockchain	`managedblockchain.` *`AWS_region`* `.amazonaws.com`
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	`kafka.`*`AWS_region`*` .amazonaws.com`
Amazon Managed Workflows for Apache Airflow (MWAA)	`airflow.`*`AWS_region`*` .amazonaws.com`
Amazon MemoryDB für Redis	`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 Amazon-Dienst	`es.`*`AWS_region`*` .amazonaws.com , aoss.`*`AWS_region`*` .amazonaws.com`
AWS Proton	`proton.`*`AWS_region`*` .amazonaws.com`
Amazon Quantum Ledger Database (Amazon QLDB)	`qldb.`*`AWS_region`*` .amazonaws.com`
Amazon RDS Performance Insights	`rds.`*`AWS_region`*` .amazonaws.com`
Amazon-Redshift	`redshift.` *`AWS_region`* `.amazonaws.com`
Amazon Redshift Query Editor V2	`sqlworkbench.` *`AWS_region`* `.amazonaws.com`

Service-Name	AWS KMS ViaService Name
Amazon Redshift Serverless	`redshift-serverless.` *`AWS_region`* `.amazonaws.com`
Amazon Rekognition	`rekognition.` *`AWS_region`* `.amazonaws.com`
Amazon Relational Database Service (Amazon RDS)	`rds.`*`AWS_region`*`.amazonaws.com`
Amazon Replicated Data Store	`ards.`*`AWS_region`*`.amazonaws.com`
Amazon SageMaker	`sagemaker.` *`AWS_region`* `.amazonaws.com`
AWS Secrets Manager	`secretsmanager.` *`AWS_region`* `.amazonaws.com`
Amazon Security Lake	`securitylake.` *`AWS_region`* `.amazonaws.com`
Amazon Simple Email Service (Amazon SES)	`ses.`*`AWS_region`*`.amazonaws.com`
Amazon Simple Notification Service (Amazon SNS)	`sns.`*`AWS_region`*`.amazonaws.com`
Amazon Simple Queue Service (Amazon SQS)	`sqs.`*`AWS_region`*`.amazonaws.com`
Amazon Simple Storage Service (Amazon S3)	`s3.`*`AWS_region`*`.amazonaws.com`
AWS Snowball	`importexport.` *`AWS_region`* `.amazonaws.com`
AWS Storage Gateway	`storagegateway.` *`AWS_region`* `.amazonaws.com`
AWS Systems Manager Incident Manager	`ssm-incidents.` *`AWS_region`* `.amazonaws.com`

Service-Name	AWS KMS ViaService Name
AWS Systems Manager Incident Manager Kontakte	`ssm-contacts.` *`AWS_region`* `.amazonaws.com`
Amazon Timestream	`timestream.` *`AWS_region`* `.amazonaws.com`
Amazon Translate	`translate.` *`AWS_region`* `.amazonaws.com`
AWS Verified Access	`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`

km: WrappingAlgorithm

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:WrappingAlgorithm`	String	Einzelwertig	`GetParametersForImport`	Schlüsselrichtlinien und IAM-Richtlinien

Dieser Bedingungsschlüssel steuert den Zugriff auf die GetParametersForImportOperation auf der Grundlage des Werts des WrappingAlgorithmParameters in der Anforderung. Sie können diese Bedingung verwenden, damit die Prinzipale während des Importvorgangs einen bestimmten Algorithmus zur Verschlüsselung von Schlüsselmaterial verwenden müssen. Anforderungen für den erforderlichen öffentlichen Schlüssel und Import-Token schlagen fehl, wenn sie einen anderen Wrapping-Algorithmus angeben.

Die folgende Beispiel-Richtlinienanweisung verwendet den Bedingungsschlüssel `kms:WrappingAlgorithm`, um dem Beispielbenutzer die Berechtigung zum Aufrufen der Produktion `GetParametersForImport` zu geben, wobei jedoch die Verwendung des Verpackungsalgorithmus `RSAES_OAEP_SHA_1` verhindert wird. Wenn der `WrappingAlgorithm` in der `GetParametersForImport`-Anforderung `RSAES_OAEP_SHA_1` lautet, schlägt die Produktion fehl.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: ExpirationModel
- km: ValidTo
- km: WrappingKeySpec

km: WrappingKeySpec

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
`kms:WrappingKeySpec`	String	Einzelwertig	`GetParametersForImport`	Schlüsselrichtlinien und IAM-Richtlinien

Dieser Bedingungsschlüssel steuert den Zugriff auf die GetParametersForImportOperation auf der Grundlage des Werts des WrappingKeySpecParameters in der Anforderung. Sie können diese Bedingung verwenden, um von den Prinzipalen zu verlangen, dass sie während des Importvorgangs einen bestimmten Typ eines öffentlichen Schlüssels verwenden. Wenn die Anforderung einen anderen Schlüsseltyp angibt, schlägt sie fehl.

Da der einzige gültige Wert für den `WrappingKeySpec`-Parameter `RSA_2048` lautet, werden die Benutzer durch Verweigern dieses Werts effektiv an der Nutzung der `GetParametersForImport`-Produktion gehindert.

Das folgende Richtlinienanweisungsbeispiel verwendet den `kms:WrappingAlgorithm`-Bedingungsschlüssel, um zu erfordern, dass der Wert für `WrappingKeySpec` in der Anforderung `RSA_4096` lautet.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

Informationen finden Sie auch unter:

- km: ExpirationModel
- km: ValidTo
- km: WrappingAlgorithm

AWS KMS Bedingungsschlüssel für AWS Nitro Enclaves

AWS Nitro Enclaves ist eine Amazon EC2 EC2-Funktion, mit der Sie isolierte Computerumgebungen, sogenannte Enklaven, erstellen können, um hochsensible Daten zu schützen und zu verarbeiten. AWS KMS bietet Bedingungsschlüssel zur Unterstützung von Nitro Enclaves. AWS Diese Bedingungsschlüssel gelten nur für Anfragen nach einer AWS KMS Nitro-Enklave.

Wenn Sie die Decrypt -, GenerateDataKeyGenerateDataKeyPair, oder GenerateRandomAPI-Operationen mit dem signierten Beglaubigungsdokument aus einer Enklave aufrufen, verschlüsseln diese APIs den Klartext in der Antwort unter dem öffentlichen Schlüssel aus dem Beglaubigungsdokument und geben Chiffretext statt Klartext zurück. Dieser Geheimtext kann nur mit dem privaten Schlüssel in der Enklave entschlüsselt werden. Weitere Informationen finden Sie unter WieAWS Nitro Enclaves AWS KMS nutzt.

Mit den folgenden Bedingungsschlüsseln können Sie die Berechtigungen für diese Operationen anhand des Inhalts des signierten Bescheinigungsdokuments einschränken. Bevor Sie einen Vorgang zulassen, AWS KMS vergleichen Sie das Bescheinigungsdokument aus der Enklave mit den Werten in diesen Bedingungsschlüsseln. AWS KMS

km: 384 RecipientAttestation ImageSha

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:RecipientAttestation:ImageSha384`	String	Einzelwertig	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
				Schlüsselrichtlinien und IAM-Richtlinien

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp
			`GenerateRandom`	

Der Bedingungsschlüssel `kms:RecipientAttestation:ImageSha384` steuert den Zugriff auf `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair`, und `GenerateRandom` mit einem KMS-Schlüssel nur dann, wenn der Image Digest aus dem signierten Bescheinigungsdokument in der Anfrage mit dem Wert im Bedingungsschlüssel übereinstimmt. Der `ImageSha384`-Wert entspricht PCR0 im Bescheinigungsdokument. Dieser Bedingungsschlüssel ist nur wirksam, wenn der `Recipient` Parameter in der Anfrage ein signiertes Bestätigungsdokument für eine AWS Nitro-Enklave angibt.

Dieser Wert ist auch in CloudTrailEreignissen für Anfragen an Nitro-Enklaven enthalten. AWS KMS

> **Note**
>
> Dieser Bedingungsschlüssel ist in Schlüsselrichtlinien-Anweisungen und IAM-Richtlinienanweisungen gültig, obwohl er nicht in der IAM-Konsole oder in der IAM-Serviceautorisierungsreferenz vorkommt.

Die folgende wichtige Richtlinienanweisung ermöglicht es der `data-processing` Rolle beispielsweise, den KMS-Schlüssel für Decrypt -, GenerateDataKey, GenerateDataKeyPair- und -Operationen zu verwenden. GenerateRandom Der Bedingungsschlüssel `kms:RecipientAttestation:ImageSha384` erlaubt die Operationen nur, wenn der Bild-Digest-Wert (PCR0) des Bescheinigungsdokuments in der Anforderung mit dem Bild-Digest-Wert in der Bedingung übereinstimmt. Dieser Bedingungsschlüssel ist nur wirksam, wenn der `Recipient` Parameter in der Anforderung ein signiertes Bestätigungsdokument für eine AWS Nitro-Enklave angibt.

Wenn die Anfrage kein gültiges Bescheinigungsdokument aus einer AWS Nitro-Enklave enthält, wird die Genehmigung verweigert, da diese Bedingung nicht erfüllt ist.

```
{
  "Sid" : "Enable enclave data processing",
```

```
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

RecipientAttestationkm: :PCR <PCR_ID>

AWS KMS Zustandsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:RecipientAttestation:PCR<PCR_ID>`	String	Einzelwertig	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:RecipientAttestation:PCR<PCR_ID>` steuert den Zugriff auf `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair`, und `GenerateRandom` mit einem KMS-Schlüssel nur dann, wenn die Platform Configuration Registers (PCRs) aus dem signierten Bescheinigungsdokument in der Anforderung mit den PCRs im Bedingungsschlüssel

übereinstimmen. Dieser Bedingungsschlüssel ist nur wirksam, wenn der `Recipient` Parameter in der Anfrage ein signiertes Bestätigungsdokument aus einer AWS Nitro-Enklave angibt.

Dieser Wert ist auch in CloudTrailEreignissen enthalten, die Anfragen an Nitro-Enklaven darstellen. AWS KMS

> **Note**
>
> Dieser Bedingungsschlüssel ist in Schlüsselrichtlinien-Anweisungen und IAM-Richtlinienanweisungen gültig, obwohl er nicht in der IAM-Konsole oder in der IAM-Serviceautorisierungsreferenz vorkommt.

Verwenden Sie das folgende Format, um einen PCR-Wert anzugeben. Verketten Sie die PCR-ID mit dem Bedingungsschlüssel-Namen. Der PCR-Wert muss eine Hexadezimalzeichenfolge in Kleinbuchstaben von bis zu 96 Bytes sein.

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

Der folgende Bedingungsschlüssel gibt beispielsweise einen bestimmten Wert für PCR1 an, der dem Hash des Kernels entspricht, der für die Enklave und den Bootstrap-Prozess verwendet wird.

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

Die folgende Beispiel-Schlüsselrichtlinienanweisung erlaubt es der`data-processing` -Rolle, den KMS-Schlüssel für die Operation Decrypt zu verwenden.

Der Bedingungsschlüssel `kms:RecipientAttestation:PCR` in dieser Anweisung erlaubt die Produktion nur, wenn der PCR1-Wert im signierten Bescheinigungsdokument in der Anforderung mit dem `kms:RecipientAttestation:PCR1`-Wert in der Bedingung übereinstimmt. Verwenden des `StringEqualsIgnoreCase`-Richtlinienoperators, um einen Vergleich der PCR-Werte ohne Berücksichtigung der Groß-/Kleinschreibung zu erfordern.

Wenn die Anforderung kein Bescheinigungsdokument enthält, wird die Berechtigung verweigert, da diese Bedingung nicht erfüllt ist.

```
{
  "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
    }
  }
}
```

ABAC für AWS KMS

Die attributbasierte Zugriffssteuerung (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. AWS KMS unterstützt ABAC, indem es Ihnen erlaubt, den Zugriff auf Ihre kundenverwaltete Schlüssel basierend auf den Tags und Aliassen steuern können, die dem KMS-Schlüsseln zugeordnet sind. Die Tag- und Alias-Bedingungsschlüssel, die ABAC in AWS KMS aktivieren, bieten eine leistungsstarke und flexible Möglichkeit, Prinzipale zur Verwendung von KMS-Schlüsseln zu autorisieren, ohne Richtlinien zu bearbeiten oder Erteilungen zu verwalten. Sie sollten diese Funktion jedoch mit Vorsicht verwenden, damit Prinzipalen der Zugriff nicht versehentlich zugelassen oder verweigert wird.

Wenn Sie ABAC verwenden, beachten Sie, dass die Berechtigung zum Verwalten von Tags und Aliassen jetzt eine Zugriffssteuerungs-Berechtigung ist. Stellen Sie sicher, dass Sie die vorhandenen Tags und Aliasse für alle KMS-Schlüssel kennen, bevor Sie eine Richtlinie bereitstellen, die von Tags oder Aliassen abhängt. Treffen Sie angemessene Vorsichtsmaßnahmen beim Hinzufügen, Löschen und Aktualisieren von Aliassen sowie beim Markieren und Entmarkieren von Schlüsseln. Erteilen Sie Berechtigungen zum Verwalten von Tags und Aliassen nur Prinzipalen, die sie benötigen, und beschränken Sie die Tags und Aliasse, die sie verwalten können.

Hinweise

Bei Verwendung von ABAC für AWS KMS, seien Sie vorsichtig, wenn Sie Prinzipalen die Berechtigung zum Verwalten von Tags und Aliassen erteilen. Wenn Sie ein Tag oder einen Alias ändern, wird die Berechtigung für einen KMS-Schlüssel eventuell erlaubt oder verweigert. Schlüsseladministratoren, die nicht über die Berechtigung zum Ändern von Schlüsselrichtlinien oder zum Erstellen von Erteilungen verfügen, können den Zugriff auf

KMS-Schlüssel steuern, wenn sie über die Berechtigung zum Verwalten von Tags oder Aliassen verfügen.
Es kann bis zu fünf Minuten dauern, bis Tag- und Alias-Änderungen Auswirkungen auf die KMS-Schlüsselautorisierung haben. Letzte Änderungen sind möglicherweise in API-Operationen sichtbar, bevor sie sich auf die Autorisierung auswirken.
Um den Zugriff auf einen KMS-Schlüssel basierend auf seinem Alias zu steuern, müssen Sie einen Bedingungsschlüssel verwenden. Sie können keinen Alias verwenden, um einen KMS-Schlüssel im `Resource`-Element einer Richtlinienanweisung darzustellen. Wenn ein Alias im `Resource`-Element erscheint, gilt die Richtlinienanweisung für den Alias und nicht für den zugeordneten KMS-Schlüssel.

Weitere Informationen

- Weitere Informationen über AWS KMS-Unterstützung für ABAC, einschließlich Beispiele, finden Sie unter Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.
- Weitere Informationen zur Verwendung von Tags, um den Zugriff auf Ihre AWS-Ressourcen zu steuern, finden Sie unter Was ist ABAC für AWS? und Steuerung des Zugriffs auf AWS-Ressourcen mit Ressourcen-Tags im IAM-Benutzerhandbuch.

ABAC-Bedingungsschlüssel für AWS KMS

Verwenden Sie die folgenden Bedingungsschlüssel in einer Schlüsselrichtlinie oder IAM-Richtlinie, um den Zugriff auf KMS-Schlüssel basierend auf deren Tags und Aliassen zu autorisieren.

ABAC-Bedingungsschlüssel	Beschreibung	Richtlinientyp	AWS KMS-Operationen
aws:ResourceTag	Tag (Schlüssel und Wert) auf dem KMS-Schlüssel entspricht dem Tag (Schlüssel und Wert) oder dem Tagmuster in der Richtlinie	Nur IAM-Richtlinie	KMS-Schlüsselressourcen-Operationen [2]

ABAC-Bedingungsschlüssel	Beschreibung	Richtlinientyp	AWS KMS-Operationen
aws:RequestTag/tag-key	Tag (Schlüssel und Wert) in der Anforderung entspricht dem Tag (Schlüssel und Wert) oder dem Tagmuster in der Richtlinie	Wichtige Richtlinien und IAM-Richtlinien [1]	TagResource, UntagResource
aws:TagKeys	Die Tag-Schlüssel in der Anforderung entsprechen den Tag-Schlüsseln in der Richtlinie	Schlüsselrichtlinien und IAM-Richtlinien [1]	TagResource, UntagResource
kms:ResourceAliases	Aliasse, die dem KMS-Schlüssel zugeordnet sind, stimmen mit den Aliassen oder Aliasmustern in der Richtlinie überein	Nur IAM-Richtlinie	KMS-Schlüsselressourcen-Operationen [2]
kms:RequestAlias	Der Alias, der den KMS-Schlüssel in der Anforderung darstellt, entspricht dem Alias oder den Aliasmustern in der Richtlinie.	Schlüsselrichtlinien und IAM-Richtlinien [1]	Kryptografische Operationen, DescribeKey, GetPublicKey

[1] Jeder Bedingungsschlüssel, der in einer Schlüsselrichtlinie verwendet werden kann, kann auch in einer IAM-Richtlinie verwendet werden, jedoch nur, wenn die Schlüsselrichtlinie es erlaubt.

[2]Eine KMS-Schlüsselressourcen-Operation ist eine Operation, die für einen bestimmten KMS-Schlüssel autorisiert ist. Um die KMS-Schlüsselressourcen-Operationen zu identifizieren, suchen

Sie in der Tabelle mit AWS KMS-Berechtigungen nach dem Wert des KMS-Schlüssels in der `Resources`-Spalte für die Operation.

Beispielsweise können Sie diese Bedingungsschlüssel verwenden, um die folgenden Richtlinien zu erstellen.

- Eine IAM-Richtlinie mit `kms:ResourceAliases`, die die Berechtigung zur Verwendung von KMS-Schlüsseln mit einem bestimmten Alias oder Aliasmuster ermöglicht. Dies unterscheidet sich etwas von Richtlinien, die auf Tags basieren: Sie können zwar Aliasmuster in einer Richtlinie verwenden, jeder Alias muss in einem AWS-Konto und einer Region jedoch eindeutig sein. Auf diese Weise können Sie eine Richtlinie auf einen ausgewählten Satz von KMS-Schlüsseln anwenden, ohne die Schlüssel-ARNs der KMS-Schlüssel in der Richtlinienanweisung aufzulisten. Um KMS-Schlüssel aus dem Satz hinzuzufügen oder zu entfernen, ändern Sie den Alias des KMS-Schlüssels.
- Eine Schlüsselrichtlinie mit `kms:RequestAlias`, die es Prinzipalen ermöglicht, einen KMS-Schlüssel in einer `Encrypt`-Operation zu nutzen, aber nur dann, wenn die `Encrypt`-Anforderung diesen Alias verwendet, um den KMS-Schlüssel zu identifizieren.
- Eine IAM-Richtlinie mit `aws:ResourceTag/`*`tag-key`*, die die Berechtigung zur Verwendung von KMS-Schlüsseln mit einem bestimmten Tag-Schlüssel und Tag-Wert verweigert. Auf diese Weise können Sie eine Richtlinie auf einen ausgewählten Satz von KMS-Schlüsseln anwenden, ohne die Schlüssel-ARNs der KMS-Schlüssel in der Richtlinienanweisung aufzulisten. Um KMS-Schlüssel aus dem Satz hinzuzufügen oder zu entfernen, markieren oder entmarkieren Sie den KMS-Schlüssel.
- Eine IAM-Richtlinie mit `aws:RequestTag/`*`tag-key`*, die es Prinzipalen erlaubt, nur `"Purpose"="Test"`-Tags von KMS-Schlüsseln zu löschen.
- Eine IAM-Richtlinie mit `aws:TagKeys`, die die Berechtigung zur Markierung oder Entmarkierung eines KMS-Schlüssels mit einem `Restricted`-Tag-Schlüssel verweigert.

ABAC macht das Zugriffsmanagement flexibel und skalierbar. Sie können beispielsweise den `aws:ResourceTag/`*`tag-key`*-Bedingungsschlüssel verwenden, um eine IAM-Richtlinie zu erstellen, die es Prinzipalen erlaubt, einen KMS-Schlüssel für bestimmte Operationen nur dann zu verwenden, wenn der KMS-Schlüssel einen `Purpose=Test`-Tag hat. Die Richtlinie gilt für alle KMS-Schlüssel in allen Regionen des AWS-Konto.

Wenn sie einem Benutzer oder einer Rolle zugeordnet ist, können Prinzipale mit der folgenden IAM-Richtlinie alle vorhandenen KMS-Schlüssel mit einem`Purpose=Test`-Tag für die angegebenen Operationen nutzen. Um diesen Zugriff auf neue oder vorhandene KMS-Schlüssel zu gewähren,

müssen Sie die Richtlinie nicht ändern. Fügen Sie einfach das `Purpose=Test`-Tag de KMS-Schlüsseln an. Um diesen Zugriff von KMS-Schlüsseln mit einem `Purpose=Test`-Tag zu entfernen, bearbeiten oder löschen Sie das 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"
        }
      }
    }
  ]
}
```

Wenn Sie diese Funktion verwenden, sollten Sie jedoch vorsichtig sein, wenn Sie Tags und Aliasse verwalten. Das Hinzufügen, Ändern oder Löschen eines Tags oder Aliasses kann versehentlich den Zugriff auf einen KMS-Schlüssel zulassen oder verweigern. Schlüsseladministratoren, die nicht über die Berechtigung zum Ändern von Schlüsselrichtlinien oder zum Erstellen von Erteilungen verfügen, können den Zugriff auf KMS-Schlüssel steuern, wenn sie über die Berechtigung zum Verwalten von Tags oder Aliassen verfügen. Um dieses Risiko zu mindern, sollten Sie Berechtigungen zum Verwalten von Tags und Aliassen beschränken. Sie können es beispielsweise nur ausgewählten Prinzipalen erlauben, `Purpose=Test`-Tags zu verwalten. Details dazu finden Sie unter Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.

Tags oder Aliasse?

AWS KMS unterstützt ABAC mit Tags und Aliassen. Beide Optionen bieten eine flexible, skalierbare Strategie zur Zugriffssteuerung, unterscheiden sich jedoch geringfügig voneinander.

Sie können sich entscheiden, Tags oder Aliasse zu verwenden, je nach Ihren speziellen AWS-Verwendungsmustern. Wenn Sie beispielsweise den meisten Administratoren bereits Markierungs-Berechtigungen erteilt haben, ist es möglicherweise einfacher, eine Autorisierungsstrategie basierend auf Aliasse zu steuern. Oder, wenn Sie nahe am Kontingent für Aliasse pro KMS-Schlüssel sind, verwenden Sie möglicherweise lieber eine Autorisierungs-Strategie, die auf Tags basiert.

Die folgenden Nutzen sind von allgemeinem Interesse.

Vorteile einer Tag-basierten Zugriffskontrolle

- Gleicher Autorisierungsmechanismus für verschiedene Arten von AWS-Ressourcen.

  Sie können denselben Tag- oder Tag-Schlüssel verwenden, um den Zugriff auf mehrere Ressourcentypen zu steuern, z. B. einen Amazon-RDS (Amazon Relational Database Service)-Cluster, ein Amazon-EBS (Amazon Elastic Block Store)-Volume und einen KMS-Schlüssel. Diese Funktion ermöglicht verschiedene Autorisierungsmodelle, die flexibler sind als herkömmliche rollenbasierte Zugriffskontrolle.
- Autorisieren des Zugriffs auf eine Gruppe von KMS-Schlüsseln

  Sie können Tags verwenden, um den Zugriff auf eine Gruppe von KMS-Schlüsseln in demselben AWS-Konto und in derselben Region zu verwalten. Weisen Sie den ausgewählten KMS-Schlüsseln denselben Tag- oder Tag-Schlüssel zu. Erstellen Sie dann eine einfache easy-to-maintain Richtlinienanweisung, die auf dem Tag oder Tag-Schlüssel basiert. Um einen KMS-Schlüssel aus Ihrer Autorisierungsgruppe hinzuzufügen oder zu entfernen, fügen Sie das Tag hinzu oder entfernen Sie es. Sie müssen die Richtlinie nicht bearbeiten.

Vorteile einer Alias-basierten Zugriffskontrolle

- Autorisieren Sie den Zugriff auf kryptografische Operationen basierend auf Aliassen.

  Die meisten anforderungsbasierten Richtlinienbedingungen für Attribute, einschließlich aws:RequestTag/tag-key, betreffen nur Operationen, die das Attribut hinzufügen, bearbeiten oder löschen. Der kms:RequestAlias-Bedingungsschlüssel steuert jedoch den Zugriff auf kryptografische Operationen basierend auf dem Alias, der zur Identifizierung des KMS-Schlüssels in der Anforderung verwendet wird. Sie können beispielsweise einem Prinzipal die Berechtigung erteilen, einen KMS-Schlüssel in einer `Encrypt`-Operation zu nutzen, aber nur dann, wenn der Wert des `KeyId`-Parameters `alias/restricted-key-1` ist. Diese Bedingung zu erfüllen, erfordert Folgendes:

- Der KMS-Schlüssel muss diesem Alias zugeordnet sein.
- Die Anforderung muss den Alias verwenden, um den KMS-Schlüssel zu identifizieren.
- Der Prinzipal muss über die Berechtigung zur Verwendung des KMS-Schlüssels verfügen, sofern die `kms:RequestAlias`-Bedingung es erlaubt.

Dies ist besonders nützlich, wenn Ihre Anwendungen häufig Aliasse oder Alias-ARNs verwenden, um auf KMS-Schlüssel zu verweisen.

- Stellen Sie sehr eingeschränkte Berechtigungen bereit.

  Ein Alias muss in einem AWS-Konto und einer Region eindeutig sein. Daher kann es wesentlich restriktiver sein, Prinzipalen Zugriff auf einen KMS-Schlüssel basierend auf einem Alias zu gewähren, als ihnen Zugriff basierend auf einem Tag zu gewähren. Im Gegensatz zu Aliassen können Tags mehreren KMS-Schlüsseln in demselben Konto und derselben Region zugewiesen werden. Wenn Sie auswählen, können Sie ein Aliasmuster verwenden, z. B.`alias/test*`, um Prinzipalen Zugriff auf eine Gruppe von KMS-Schlüsseln in demselben Konto und derselben Region zu gewähren. Allerdings ermöglicht das Erlauben oder Verweigern des Zugriffs auf einen bestimmten Alias eine sehr strenge Kontrolle über KMS-Schlüssel.

Fehlerbehebung bei ABAC für AWS KMS

Die Steuerung des Zugriffs auf KMS-Schlüssel basierend auf ihren Tags und Aliassen ist bequem und leistungsstark. Es ist jedoch anfällig für einige vorhersehbare Fehler, die Sie verhindern möchten.

Zugriff aufgrund von Tag-Änderung geändert

Wenn ein Tag gelöscht wird oder sein Wert geändert wird, wird Prinzipalen, dessen Zugriff auf einen KMS-Schlüssel nur auf diesem Tag basiert, der Zugriff auf den KMS-Schlüssel verweigert. Dies kann auch passieren, wenn ein Tag, das in einer Zugriffsverweigerungs-Richtlinienanweisung enthalten ist, einem KMS-Schlüssel hinzugefügt wird. Das Hinzufügen eines richtlinienbezogenen Tags zu einem KMS-Schlüssel kann Prinzipalen den Zugriff erlauben, denen der Zugriff auf einen KMS-Schlüssel verweigert werden soll.

Angenommen, ein Prinzipal hat Zugriff auf einen KMS-Schlüssel basierend auf dem `Project=Alpha`-Tag, z. B. die Berechtigung, die in der folgenden IAM-Richtlinienanweisung bereitgestellt wird.

```
{
```

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

Wenn das Tag aus diesem KMS-Schlüssel gelöscht wird oder der Tag-Wert geändert wird, hat der Prinzipal keine Berechtigung mehr, den KMS-Schlüssel für die angegebenen Operationen zu verwenden. Dies kann offensichtlich werden, wenn der Prinzipal versucht, Daten in einem - AWSService zu lesen oder zu schreiben, der einen vom Kunden verwalteten Schlüssel verwendet. Um die Tag-Änderung zu verfolgen, überprüfen Sie Ihre CloudTrail Protokolle auf - TagResource oder UntagResource -Einträge.

Um den Zugriff wiederherzustellen, ohne die Richtlinie zu aktualisieren, ändern Sie die Tags auf dem KMS-Schlüssel. Diese Aktion hat minimale Auswirkungen außer für einen kurzen Zeitraum, während sie in allen Bereichen von AWS KMS in Kraft tritt. Um einen Fehler wie diesen zu vermeiden, erteilen Sie Berechtigungen zum Markieren und Entmarkieren nur an Prinzipale, die sie benötigen, und beschränken Sie ihrer Markierungs-Berechtigungen auf Tags, die sie verwalten müssen. Bevor Sie einen Tag ändern, durchsuchen Sie Richtlinien, um den Zugriff zu erkennen, der vom Tag abhängt, und erhalten Sie KMS-Schlüssel in allen Regionen, die das Tag enthalten. Sie können erwägen, einen Amazon- CloudWatch Alarm zu erstellen, wenn bestimmte Tags geändert werden.

Änderung des Zugriffs aufgrund von Aliasänderungen

Wenn ein Alias gelöscht oder einem anderen KMS-Schlüssel zugeordnet wird, wird Prinzipalen, dessen Zugriff auf den KMS-Schlüssel nur auf diesem Alias basiert, der Zugriff auf den KMS-Schlüssel verweigert. Dies kann auch passieren, wenn ein Tag, das in einer Zugriffsverweigerungs-Richtlinienanweisung enthalten ist, einem KMS-Schlüssel hinzugefügt wird. Das Hinzufügen eines

richtlinienbezogenen Tags zu einem KMS-Schlüssel kann Prinzipalen den Zugriff erlauben, denen der Zugriff auf einen KMS-Schlüssel verweigert werden soll.

Die folgende IAM-Richtlinienanweisung verwendet beispielsweise den kms:ResourceAliases-Bedingungsschlüssel, um den Zugriff auf KMS-Schlüssel in verschiedenen Regionen des Kontos mit einem der angegebenen Aliasse zu ermöglichen.

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

Um die Aliasänderung nachzuverfolgen, überprüfen Sie Ihre - CloudTrail Protokolle auf CreateAlias-UpdateAlias, - und -DeleteAliasEinträge.

Um den Zugriff wiederherzustellen, ohne die Richtlinie zu aktualisieren, ändern Sie den Alias, der dem KMS-Schlüssel zugeordnet ist. Da jeder Alias nur einem KMS-Schlüssel in einem Konto und einer Region zugeordnet werden kann, ist die Verwaltung von Aliassen etwas schwieriger als die Verwaltung von Tags. Das Wiederherstellen des Zugriffs auf einige Prinzipale auf einem KMS-Schlüssel kann denselben oder anderen Prinzipalen den Zugriff auf einen anderen KMS-Schlüssel verweigern.

Um diesen Fehler zu vermeiden, erteilen Sie Alias-Verwaltungs-Berechtigungen nur für Prinzipale, die diese benötigen, und schränken Sie ihre Alias-Verwaltungs-Berechtigungen auf Aliasse ein, die sie verwalten müssen. Suchen Sie vor dem Aktualisieren oder Löschen eines Aliasses Richtlinien, um den Zugriff zu erkennen, der vom Alias abhängt, und finden Sie die KMS-Schlüssel in allen Regionen, die dem Alias zugeordnet sind.

Zugriff aufgrund eines Aliaskontingents verweigert

Benutzer, die von einer kms:ResourceAliases-Bedingung zur Verwendung eines KMS-Schlüssels autorisiert sind, erhalten eine `AccessDenied` Ausnahme, wenn der KMS-Schlüssel das Standardkontingent für Aliase pro KMS-Schlüssel für dieses Konto und diese Region überschreitet.

Um den Zugriff wiederherzustellen, löschen Sie Aliasse, die dem KMS-Schlüssel zugeordnet sind, damit er dem Kontingent entspricht. Oder verwenden Sie einen alternativen Mechanismus, um Benutzern Zugriff auf den KMS-Schlüssel zu gewähren.

Verzögerte Autorisierungsänderung

Es kann bis zu fünf Minuten dauern, bis sich die Änderungen, die Sie an Tags und Aliassen vornehmen, auf die Autorisierung von KMS-Schlüsseln auswirken. Infolgedessen kann eine Tag- oder Aliasänderung in den Antworten von API-Operationen widergespiegelt werden, bevor sie sich auf die Autorisierung auswirken. Diese Verzögerung ist wahrscheinlich länger als die kurze eventuelle Konsistenzverzögerung, die die meisten AWS KMS-Operationen betrifft.

Beispielsweise können Sie eine IAM-Richtlinie verwenden, die es bestimmten Prinzipalen erlaubt, einen KMS-Schlüssel mit einem `"Purpose"="Test"`-Tag zu nutzen. Dann fügen Sie das `"Purpose"="Test"`-Tag einem KMS-Schlüssel hinzu. Obwohl der TagResource Vorgang abgeschlossen ist und die ListResourceTags Antwort bestätigt, dass das Tag dem KMS-Schlüssel zugewiesen ist, haben die Prinzipale möglicherweise bis zu fünf Minuten lang keinen Zugriff auf den KMS-Schlüssel.

Um Fehler zu vermeiden, bauen Sie diese erwartete Verzögerung in Ihren Code ein.

Fehlgeschlagene Anforderungen aufgrund von Alias-Aktualisierungen

Sie können Aliasse auch aktualisieren, wodurch ein vorhandenes Alias einem anderen KMS-Schlüssel zugeordnet wird.

Die Entschlüsselung von - und -ReEncryptAnforderungen, die den Aliasnamen oder Alias-ARN angeben, schlägt möglicherweise fehl, da der Alias jetzt einem KMS-Schlüssel zugeordnet ist, der den Chiffretext nicht verschlüsselt hat. Diese Situation gibt in der Regel `IncorrectKeyException`

oder `NotFoundException` zurück. Oder wenn die Anforderung keinen `KeyId`- oder `DestinationKeyId`-Parameter enthält, schlägt die Operation möglicherweise mit einer `AccessDenied`-Ausnahme fehl, da der Aufrufer keinen Zugriff mehr auf den KMS-Schlüssel hat, der den Chiffretext verschlüsselt hat.

Sie können die Änderung verfolgenCreateAlias, indem Sie sich die CloudTrail Protokolle für UpdateAlias-, - und -DeleteAliasProtokolleinträge ansehen. Sie können auch den Wert des `LastUpdatedDate` Felds in der ListAliases Antwort verwenden, um eine Änderung zu erkennen.

Die folgende ListAliases Beispielantwort zeigt beispielsweise, dass der `ProjectAlpha_Test` Alias in der `kms:ResourceAliases` Bedingung aktualisiert wurde. Daher verlieren die Prinzipale, dessen Zugriff auf dem Alias basiert, den Zugriff auf den zuvor zugeordneten KMS-Schlüssel. Stattdessen haben sie Zugriff auf den neu zugeordneten KMS-Schlüssel.

```
$ 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
        }
    ]
}
```

Die Lösung für diese Änderung ist nicht einfach. Sie können den Alias erneut aktualisieren, um ihn dem ursprünglichen KMS-Schlüssel zuzuordnen. Bevor Sie jedoch handeln, müssen Sie die Auswirkungen dieser Änderung auf den aktuell zugeordneten KMS-Schlüssel berücksichtigen. Wenn Prinzipale den letzteren KMS-Schlüssel in kryptografischen Operationen verwendet haben, müssen sie möglicherweise weiterhin darauf zugreifen. In diesem Fall sollten Sie die Richtlinie aktualisieren,

um sicherzustellen, dass Prinzipale über die Berechtigung verfügen, beide KMS-Schlüssel zu verwenden.

Sie können einen Fehler wie diesen verhindern: Bevor Sie einen Alias aktualisieren, suchen Sie Richtlinien, um den Zugriff zu erkennen, der vom Alias abhängt. Rufen Sie dann KMS-Schlüssel in allen Regionen ab, die dem Alias zugeordnet sind. Erteilen Sie Alias-Verwaltungs-Berechtigungen nur für Prinzipale, die diese benötigen, und schränken Sie ihre Alias-Verwaltungs-Berechtigungen auf Aliasse ein, die sie verwalten müssen.

Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben

Sie können IAM-Benutzern oder -Rollen in einem anderen AWS-Konto die Verwendung eines KMS-Schlüssels in Ihrem Konto erlauben. Der kontoübergreifende Zugriff erfordert die Berechtigung in der Schlüsselrichtlinie des KMS-Schlüssels und in einer IAM-Richtlinie im Konto des externen Benutzers.

Die kontenübergreifende Berechtigung gilt nur für die folgenden Operationen:

- Kryptografische Operationen
- CreateGrant
- DescribeKey
- GetKeyRotationStatus
- GetPublicKey
- ListGrants
- RetireGrant
- RevokeGrant

Wenn Sie einem Benutzer in einem anderen Konto die Berechtigung für andere Operationen erteilen, haben diese Berechtigungen keine Auswirkungen. Wenn Sie beispielsweise einem Prinzipal in einem anderen Konto kms:ListKeys Berechtigung in einer IAM-Richtlinie oder kms:ScheduleKeyDeletion Berechtigung für einen KMS-Schlüssel in einer Schlüsselrichtlinie erteilen, schlagen die Versuche des Benutzers, diese Operationen für Ihre Ressourcen aufzurufen, weiterhin fehl.

Weitere Informationen zur Verwendung von KMS-Schlüsseln in verschiedenen Konten für AWS KMS-Operationen finden Sie in der Spalte Kontenübergreifende Nutzung in der AWS KMS Berechtigungen und Verwenden von KMS-Schlüsseln in anderen Konten. Es gibt auch einen Abschnitt über

kontenübergreifende Nutzung in jeder API-Beschreibung in der AWS Key Management Service-API-Referenz.

> **Warning**
>
> Seien Sie vorsichtig, wenn Sie Prinzipalen Berechtigungen zur Verwendung Ihrer KMS-Schlüssel erteilen. Wenn möglich, befolgen Sie dem Prinzip der geringsten Berechtigung. Geben Sie Benutzern nur Zugriff auf die KMS-Schlüssel, die sie für die erforderlichen Vorgänge benötigen.
> Seien Sie auch vorsichtig bei der Verwendung eines unbekannten KMS-Schlüssels, insbesondere eines KMS-Schlüssels in einem anderen Konto. Bösartige Benutzer geben Ihnen möglicherweise die Berechtigung, ihren KMS-Schlüssel zu verwenden, um Informationen über Sie oder Ihr Konto abzurufen.
> Weitere Informationen zur Verwendung von Richtlinien zum Steuern des Zugriffs auf die Ressourcen in Ihrem Konto finden Sie unter Bewährte Methoden für IAM-Richtlinien.

Um Benutzern und Rollen in einem anderen Konto die Berechtigung zur Verwendung eines KMS-Schlüssels zu erteilen, müssen Sie zwei verschiedene Arten von Richtlinien verwenden:

- Die Schlüsselrichtlinie für den KMS-Schlüssel muss dem externen Konto (oder den Benutzern und Rollen im externen Konto) die Berechtigung zur Verwendung des KMS-Schlüssels erteilen. Die Schlüsselrichtlinie ist in dem Konto definiert, in dem der KMS-Schlüssel hinterlegt ist.
- IAM-Richtlinien im externen Konto müssen die Schlüsselrichtlinien-Berechtigungen an seine Benutzer und Rollen delegieren. Diese Richtlinien werden im externen Konto festgelegt und erteilen Berechtigungen für Benutzer und Rollen in diesem Konto.

Die Schlüsselrichtlinie bestimmt, wer Zugriff auf den KMS-Schlüssel haben kann. Die IAM-Richtlinie bestimmt, wer Zugriff auf den KMS-Schlüssel hat. Weder die Schlüsselrichtlinie noch die IAM-Richtlinie allein sind ausreichend – Sie müssen beides ändern.

Um die Schlüsselrichtlinie zu bearbeiten, können Sie die Richtlinienansicht in der AWS Management Console oder die CreateKey- oder -PutKeyPolicyOperationen verwenden. Weitere Informationen zum Festlegen der Schlüsselrichtlinie beim Erstellen eines KMS-Schlüssels finden Sie unter Erstellen von KMS-Schlüssel, die von anderen Konten verwendet werden können.

Hilfe zum Bearbeiten von IAM-Richtlinien finden Sie unter Verwenden von IAM-Richtlinien mit AWS KMS.

Ein Beispiel, das zeigt, wie die Schlüsselrichtlinie und IAM-Richtlinien zusammenarbeiten, um die Verwendung eines KMS-Schlüssel in einem anderen Konto zu ermöglichen, finden Sie unter Beispiel 2: Der Benutzer übernimmt eine Rolle mit der Berechtigung zur Verwendung eines KMS-Schlüssels in einem anderen AWS-Konto..

Sie können die resultierenden kontenübergreifenden AWS KMS-Operationen auf dem KMS-Schlüssel in Ihren AWS CloudTrail-Protokollen anzeigen. Operationen, die KMS-Schlüssel in anderen Konten verwenden, werden sowohl im Konto des Anrufers als auch im Konto des KMS-Schlüsselbesitzers protokolliert.

Themen

> **Note**
>
> Die Beispiele in diesem Thema zeigen, wie Sie eine Schlüsselrichtlinie und eine IAM-Richtlinie zusammen verwenden, um den Zugriff auf einen KMS-Schlüssel bereitzustellen und einzuschränken. Diese generischen Beispiele sollen nicht die Berechtigungen darstellen, die ein bestimmter AWS-Service von einem KMS-Schlüssel erfordert. Weitere Informationen zu Berechtigungen, die ein AWS-Service benötigt, finden Sie im Thema zur Verschlüsselung in der Service-Dokumentation.

Schritt 1: Hinzufügen einer Schlüsselrichtlinienanweisung im lokalen Konto

Die Schlüsselrichtlinie für einen KMS-Schlüssel ist der erste Bestimmungsfaktor dafür, wer auf den KMS-Schlüssel zugreifen kann und welche Operationen sie ausführen können. Die Schlüsselrichtlinie ist immer in dem Konto, in dem der KMS-Schlüssel hinterlegt ist. Anders als IAM-Richtlinien geben Schlüsselrichtlinien keine Ressource an. Die Ressource ist der KMS-Schlüssel, der die Schlüsselrichtlinie zugeordnet ist. Wenn Sie eine kontenübergreifende Berechtigung erteilen, muss die Schlüsselrichtlinie für den KMS-Schlüssel dem externen Konto (oder den Benutzern und Rollen im externen Konto) die Berechtigung erteilen, den KMS-Schlüssel zu verwenden.

Um einem externen Konto die Berechtigung zur Verwendung des KMS-Schlüssels zu erteilen, fügen Sie der Schlüsselrichtlinie eine Anweisung hinzu, die das externe Konto angibt. Geben Sie im `Principal`-Element der Schlüsselrichtlinie den Amazon-Ressourcennamen (ARN) des externen Kontos ein.

Wenn Sie ein externes Konto in einer Schlüsselrichtlinie angeben, können IAM-Administratoren im externen Konto IAM-Richtlinien verwenden, um diese Berechtigungen an alle Benutzer und Rollen im externen Konto zu delegieren. Sie können auch entscheiden, welche der in der Schlüsselrichtlinie angegebenen Aktionen die Benutzer und Rollen ausführen dürfen.

Berechtigungen, die dem externen Konto und seinen Prinzipalen erteilt werden, sind nur wirksam, wenn das externe Konto in der Region aktiviert ist, die den KMS-Schlüssel und seine Schlüsselrichtlinie hostet. Informationen zu Regionen, die standardmäßig nicht aktiviert sind („Opt-In-Regionen“), finden Sie unter Verwalten von AWS-Regionen in Allgemeine AWS-Referenz.

Angenommen, Sie möchten Konto `444455556666` erlauben, einen symmetrischen KMS-Schlüssel im Konto `111122223333` zu verwenden. Fügen Sie dazu der Schlüsselrichtlinie für den KMS-Schlüssel in Konto `111122223333` eine Schlüsselrichtlinie wie die im folgenden Beispiel hinzu. Diese Richtlinienanweisung erteilt dem externen Konto `444455556666` die Berechtigung, den KMS-Schlüssel in kryptografischen Operationen für KMS-Schlüssel mit symmetrischer Verschlüsselung zu verwenden.

> **Note**
>
> Das folgende Beispiel stellt ein Beispiel für eine Schlüsselrichtlinie für die gemeinsame Nutzung eines KMS-Schlüssels mit einem anderen Konto dar. Ersetzen Sie die Werte `Sid`, `Principal` und `Action` im Beispiel durch gültige Werte für die vorgesehene Verwendung Ihres KMS-Schlüssels.

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

Anstatt dem externen Konto die Berechtigung zu erteilen, können Sie bestimmte externe Benutzer und Rollen in der Schlüsselrichtlinie angeben. Diese Benutzer und Rollen können den KMS-Schlüssel jedoch erst verwenden, wenn IAM-Administratoren im externen Konto ihren Identitäten die richtigen IAM-Richtlinien anfügen. Die IAM-Richtlinien können allen oder einer Teilmenge der externen Benutzer und Rollen, die in der Schlüsselrichtlinie angegeben sind, die Berechtigung erteilen. Und sie können alle oder eine Teilmenge der Aktionen zulassen, die in der Schlüsselrichtlinie angegeben sind.

Die Angabe von Identitäten in einer Schlüsselrichtlinie beschränkt die Berechtigungen, die IAM-Administratoren im externen Konto bereitstellen können. Die Richtlinienverwaltung mit zwei Konten wird jedoch komplexer. Angenommen, Sie müssen einen Benutzer oder eine Rolle hinzufügen. Sie müssen diese Identität der Schlüsselrichtlinie in dem Konto hinzufügen, das den KMS-Schlüssel besitzt, und IAM-Richtlinien im Konto der Identität erstellen.

Um bestimmte externe Benutzer oder Rollen in einer Schlüsselrichtlinie anzugeben, geben Sie im `Principal`-Element den Amazon-Ressourcennamen (ARN) eines Benutzers oder einer Rolle im externen Konto ein.

Die folgende Schlüsselrichtlinienanweisung erlaubt es beispielsweise `ExampleRole` im Konto `444455556666`, einen KMS-Schlüssel im Konto `111122223333` zu verwenden. Diese Schlüssel-Richtlinienanweisung erteilt dem externen Konto `444455556666` die Berechtigung, den KMS-Schlüssel in kryptografischen Operationen für KMS-Schlüssel mit symmetrischer Verschlüsselung zu verwenden.

> **Note**
>
> Das folgende Beispiel stellt ein Beispiel für eine Schlüsselrichtlinie für die gemeinsame Nutzung eines KMS-Schlüssels mit einem anderen Konto dar. Ersetzen Sie die Werte `Sid`, `Principal` und `Action` im Beispiel durch gültige Werte für die vorgesehene Verwendung Ihres KMS-Schlüssels.

```
{
    "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**
>
> Setzen Sie den Prinzipal nicht auf ein Sternchen (*) in einer Schlüsselrichtlinienanweisung, die Berechtigungen erlaubt, es sei denn, Sie verwenden Bedingungen, um die Schlüsselrichtlinie einzuschränken. Ein Sternchen gibt jeder Identität in jedem AWS-Konto die Berechtigung, den KMS-Schlüssel zu verwenden, es sei denn, eine andere Richtlinienanweisung verweigert dies explizit. Benutzer in anderen AWS-Konten können Ihren KMS-Schlüssel verwenden, wenn sie in ihrem eigenen Konto über entsprechende Berechtigungen verfügen.

Sie müssen auch entscheiden, welche Berechtigungen Sie dem externen Konto erteilen möchten. Eine Liste der Berechtigungen für KMS-Schlüssel finden Sie unter AWS KMS Berechtigungen.

Sie können dem externen Konto die Berechtigung erteilen, den KMS-Schlüssel in kryptografischen Operationen zu verwenden und den KMS-Schlüssel mit AWS-Services zu verwenden, die in AWS KMS integriert sind. Verwenden Sie dazu den Abschnitt Key Users (Schlüsselbenutzer) der AWS Management Console. Details hierzu finden Sie unter Erstellen von KMS-Schlüssel, die von anderen Konten verwendet werden können.

Um andere Berechtigungen in Schlüsselrichtlinien anzugeben, bearbeiten Sie das Schlüsselrichtliniendokument. Sie können beispielsweise Benutzern die Berechtigung zum Entschlüsseln, aber nicht zum Verschlüsseln erteilen, oder die Berechtigung zum Anzeigen des KMS-Schlüssels, aber nicht zur Verwendung. Um das Schlüsselrichtliniendokument zu bearbeiten,

können Sie die Richtlinienansicht in der AWS Management Console oder den CreateKey - oder -PutKeyPolicyOperationen verwenden.

Schritt 2: Hinzufügen von IAM-Richtlinien im externen Konto

Die Schlüsselrichtlinie im Konto, dem der KMS-Schlüssel gehört, legt den gültigen Bereich für Berechtigungen fest. Benutzer und Rollen im externen Konto können den KMS-Schlüssel erst verwenden, wenn Sie IAM-Richtlinien anfügen, die diese Berechtigungen delegieren, oder Erteilungen verwenden, um den Zugriff auf den KMS-Schlüssel zu verwalten. Die IAM-Richtlinien werden im externen Konto festgelegt.

Wenn die Schlüsselrichtlinie dem externen Konto die Berechtigung erteilt, können Sie jedem Benutzer oder jeder Rolle im Konto IAM-Richtlinien zuweisen. Aber wenn die Schlüsselrichtlinie den angegebenen Benutzern oder Rollen die Berechtigung erteilt, kann die IAM-Richtlinie diese Berechtigungen nur allen oder einer Teilmenge der angegebenen Benutzer und Rollen erteilen. Wenn eine IAM-Richtlinie anderen externen Benutzern oder Rollen den Zugriff auf KMS-Schlüssel gewährt, hat dies keine Auswirkungen.

Die Schlüsselrichtlinie beschränkt auch die Aktionen in der IAM-Richtlinie. Die IAM-Richtlinie kann alle oder eine Teilmenge der Aktionen delegieren, die in der Schlüsselrichtlinie angegeben sind. Wenn die IAM-Richtlinie Aktionen auflistet, die nicht in der Schlüsselrichtlinie angegeben sind, sind diese Berechtigungen nicht wirksam.

Die folgende IAM-Beispielrichtlinie erlaubt es dem Prinzipal, den KMS-Schlüssel im Konto `111122223333` für kryptografische Operationen zu verwenden. Um diese Berechtigung Benutzern und Rollen im Konto `444455556666` zu erteilen, fügen Sie die Richtlinie den Benutzern oder Rollen im Konto `444455556666` an.

> **Note**
>
> Das folgende Beispiel stellt eine IAM-Richtlinie für die gemeinsame Nutzung eines KMS-Schlüssels mit einem anderen Konto dar. Ersetzen Sie die Werte `Sid`, `Resource` und `Action` im Beispiel durch gültige Werte für die vorgesehene Verwendung Ihres KMS-Schlüssels.

```
{
    "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"
}
```

Beachten Sie die folgenden Details über diese Richtlinie:

- Im Gegensatz zu Schlüsselrichtlinien enthalten IAM-Richtlinienanweisungen das `Principal`-Element nicht. In IAM-Richtlinien ist der Prinzipal die Identität, der die Richtlinie angefügt ist.
- Das `Resource`-Element in der IAM-Richtlinie identifiziert den KMS-Schlüssel, den der Prinzipal verwenden kann. Um einen KMS-Schlüssel anzugeben, fügen Sie dessen Schüssel-ARN dem `Resource`-Element hinzu.
- Sie können mehrere KMS-Schlüssel im `Resource`-Element angeben. Wenn Sie jedoch keine bestimmten KMS-Schlüssel im `Resource`-Element angeben, können Sie versehentlich Zugriff auf mehr KMS-Schlüssel gewähren, als Sie beabsichtigen.
- Damit der externe Benutzer den KMS-Schlüssel mit AWS-Services verwenden kann, die in AWS KMS integriert sind, müssen Sie möglicherweise Berechtigungen zur Schlüsselrichtlinie oder der IAM-Richtlinie hinzufügen. Details hierzu finden Sie unter Zulassen der Verwendung externer KMS-Schlüssel mit AWS-Services.

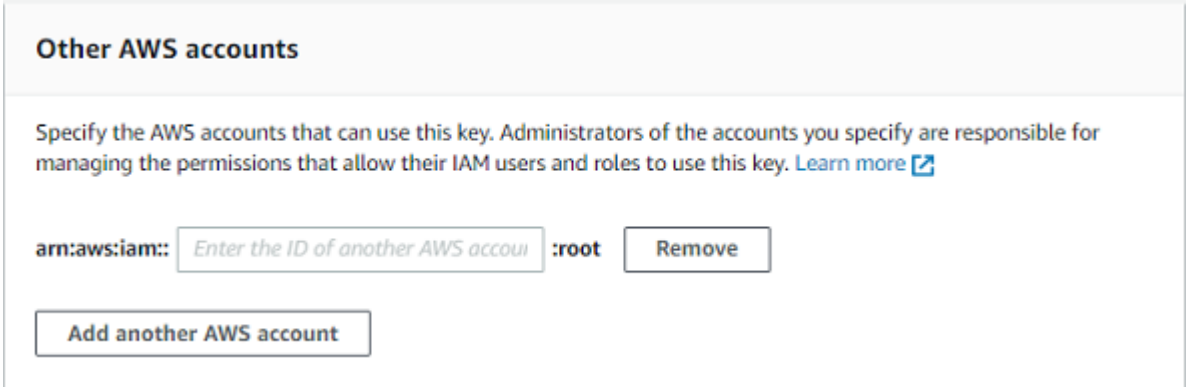
Weitere Informationen zur Arbeit mit IAM-Richtlinien finden Sie unter IAM-Richtlinien.

Erstellen von KMS-Schlüssel, die von anderen Konten verwendet werden können

Wenn Sie die -CreateKeyOperation verwenden, um einen KMS-Schlüssel zu erstellen, können Sie seinen `Policy` Parameter verwenden, um eine Schlüsselrichtlinie anzugeben, die einem externen Konto oder externen Benutzern und Rollen die Berechtigung zur Verwendung des KMS-Schlüssels erteilt. Sie müssen auch IAM-Richtlinien im externen Konto hinzufügen, die diese Berechtigungen an die Benutzer und Rollen des Kontos delegieren, auch wenn Benutzer und Rollen in der Schlüsselrichtlinie angegeben sind. Sie können die Schlüsselrichtlinie jederzeit ändern, indem Sie die -PutKeyPolicyOperation verwenden.

Wenn Sie einen KMS-Schlüssel in der AWS Management Console erstellen, erstellen Sie auch dessen Schlüsselrichtlinie. Wenn Sie Identitäten im Abschnitt Key Administrators (Schlüsseladministratoren) und Key Users (Schlüsselbenutzer) auswählen, fügt AWS KMS Richtlinienanweisungen für diese Identitäten zur Schlüsselrichtlinie des KMS-Schlüssels hinzu.

Im Abschnitt Key Users (Schlüsselbenutzer) können Sie auch externe Konten als Schlüsselbenutzer hinzufügen.



Wenn Sie die Konto-ID eines externen Kontos eingeben, fügt AWS KMS der Schlüsselrichtlinie zwei Anweisungen hinzu. Diese Aktion wirkt sich nur auf die Schlüsselrichtlinie aus. Benutzer und Rollen im externen Konto können den KMS-Schlüssel erst verwenden, wenn Sie IAM-Richtlinien anfügen, um ihnen einige oder alle dieser Berechtigungen zu erteilen.

Die erste Schlüsselrichtlinienanweisung erteilt dem externen Konto die Berechtigung, den KMS-Schlüssel in kryptografischen Operationen zu verwenden.

> **Note**
>
> Die folgenden Beispiele stellen eine Schlüsselrichtlinie für die gemeinsame Nutzung eines KMS-Schlüssels mit einem anderen Konto dar. Ersetzen Sie die Werte `Sid`, `Principal` und `Action` im Beispiel durch gültige Werte für die vorgesehene Verwendung Ihres KMS-Schlüssels.

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
```

```
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Die zweite Schlüsselrichtlinienanweisung erlaubt es dem externen Konto, Erteilungen für den KMS-Schlüssel zu erstellen, anzuzeigen und zu widerrufen, aber nur, wenn die Anforderung von einem AWS-Service stammt, der in AWS KMS integriert ist. Diese Berechtigungen erlauben es anderen AWS-Services, die Benutzerdaten verschlüsseln, den KMS-Schlüssel zu verwenden.

Diese Berechtigungen sind für KMS-Schlüssel konzipiert, die Benutzerdaten in -AWSServices wie Amazon WorkMail verschlüsseln. Diese Services verwenden in der Regel Erteilungen, um die Berechtigungen zu erhalten, die sie für die Verwendung des KMS-Schlüssels im Namen des Benutzers benötigen. Details hierzu finden Sie unter Zulassen der Verwendung externer KMS-Schlüssel mit AWS-Services.

```
{
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": "true"
        }
    }
}
```

Wenn diese Berechtigungen nicht Ihren Anforderungen entsprechen, können Sie sie in der Konsolenrichtlinienansicht oder mithilfe der PutKeyPolicy Operation bearbeiten. Sie können

bestimmte externe Benutzer und Rollen angeben, anstatt dem externen Konto die Berechtigung zu erteilen. Sie können die Aktionen ändern, die die Richtlinie angibt. Außerdem können Sie globale und AWS KMS-Richtlinienbedingungen verwenden, um die Berechtigungen einzuschränken.

Zulassen der Verwendung externer KMS-Schlüssel mit AWS-Services

Sie können einem Benutzer in einem anderen Konto die Berechtigung erteilen, Ihren KMS-Schlüssel mit einem Service zu verwenden, der in AWS KMS integriert ist. Beispielsweise kann ein Benutzer in einem externen Konto Ihren KMS-Schlüssel verwenden, um die Objekte in einem Amazon-S3-Bucket zu verschlüsseln oder die in AWS Secrets Manager gespeicherten Geheimnisse zu verschlüsseln.

Die Schlüsselrichtlinie muss dem externen Benutzer oder dem Konto des externen Benutzers die Berechtigung zur Verwendung des KMS-Schlüssels erteilen. Darüber hinaus müssen Sie IAM-Richtlinien an die Identität anfügen, die dem Benutzer die Berechtigung zur Nutzung des AWS-Service erteilt. Außerdem erfordert der Service möglicherweise, dass Benutzer über zusätzliche Berechtigungen in der Schlüsselrichtlinie oder IAM-Richtlinie verfügen. Eine Liste der Berechtigungen, die der AWS-Service für einen vom Kunden verwalteten Schlüssel benötigt, finden Sie im Thema „Datenschutz“ im Kapitel „Sicherheit“ des Benutzerhandbuchs oder des Entwicklerhandbuchs für den Service.

Verwenden von KMS-Schlüsseln in anderen Konten

Wenn Sie über die Berechtigung verfügen, einen KMS-Schlüssel in einem anderen AWS-Konto zu verwenden, können Sie den KMS-Schlüssel in der AWS Management Console, den AWS-SDKs, der AWS CLI und AWS Tools for PowerShell verwenden.

Um einen KMS-Schlüssel in einem anderen Konto in einem Shell-Befehl oder einer API-Anforderung zu identifizieren, verwenden Sie die folgenden Schlüsselbezeichner.

- GetPublicKeyVerwenden Sie für kryptografische Operationen , DescribeKeyund den Schlüssel-ARN oder Alias-ARN des KMS-Schlüssels.
- RevokeGrantVerwenden Sie für CreateGrant, GetKeyRotationStatusListGrants, und den Schlüssel-ARN des KMS-Schlüssels.

Wenn Sie nur eine Schlüssel-ID oder einen Aliasnamen eingeben, geht AWS davon aus, dass sich der KMS-Schlüssel in Ihrem Konto befindet.

Die AWS KMS-Konsole zeigt keine KMS-Schlüssel in anderen Konten an, selbst wenn Sie über die Berechtigung verfügen, sie zu verwenden. Außerdem enthalten die Listen der KMS-Schlüssel, die in den Konsolen anderer AWS-Services angezeigt werden, keine KMS-Schlüssel in anderen Konten.

Um einen KMS-Schlüssel in einem anderen Konto in der Konsole eines AWS-Service anzugeben, müssen Sie den Schlüssel-ARN oder Alias-ARN des KMS-Schlüssels eingeben. Die erforderliche Schlüssel-ID variiert je nach Service und kann auch zwischen der Servicekonsole und ihren API-Operationen variieren. Weitere Informationen finden Sie in der Service-Dokumentation.

Verwenden von serviceverknüpften Rollen für AWS KMS

AWS Key Management Service verwendet serviceverknüpfte Rollen von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit AWS KMS verknüpft ist. Serviceverknüpfte Rollen werden von AWS KMSdefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von AWS KMS, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS KMS definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur AWS KMS die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Dies schützt Ihre AWS KMS-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für AWS KMS benutzerdefinierte Schlüsselspeicher

AWS KMS verwendet eine serviceverknüpfte Rolle namens AWSServiceRoleForKeyManagementServiceCustomKeyStores, um benutzerdefinierte Schlüsselspeicher zu unterstützen. Diese serviceverknüpfte Rolle gibt AWS KMS die Berechtigung,

Ihre AWS CloudHSM-Cluster anzuzeigen und die Netzwerkinfrastruktur zu erstellen, die eine Verbindung zwischen Ihrem benutzerdefinierten Schlüsselspeicher und dessen AWS CloudHSM-Cluster unterstützt. AWS KMS erstellt diese Rolle nur, wenn Sie einen benutzerdefinierten Schlüsselspeicher erstellen. Sie können diese serviceverknüpfte Rolle nicht direkt erstellen.

Die serviceverknüpfte Rolle AWSServiceRoleForKeyManagementServiceCustomKeyStores vertraut darauf, dass `cks.kms.amazonaws.com` die Rolle annimmt. Dies hat zur Folge, dass nur AWS KMS diese serviceverknüpfte Rolle annehmen kann.

Die Berechtigungen in der Rolle werden auf die Aktionen beschränkt, die AWS KMS ausführt, um eine Verbindung zwischen einem benutzerdefinierten Schlüsselspeicher und einem AWS CloudHSM-Cluster herzustellen. AWS KMS werden keine weiteren Berechtigungen erteilt. Beispielsweise ist AWS KMS nicht zum Erstellen, Verwalten oder Löschen Ihrer AWS CloudHSM-Cluster, HSMs oder Sicherungen berechtigt.

Weitere Informationen zur AWSServiceRoleForKeyManagementServiceCustomKeyStores-Rolle, einschließlich einer Liste der Berechtigungen und Anweisungen zum Anzeigen der Rolle, Bearbeiten der Rollenbeschreibung, Löschen der Rolle und Neuerstellen mithilfe von AWS KMS finden Sie unter Autorisieren von AWS KMS für die Verwaltung von AWS CloudHSM- und Amazon-EC2-Ressourcen.

Berechtigungen von serviceverknüpften Rollen für multiregionale AWS KMS-Schlüssel

AWS KMS verwendet eine serviceverknüpfte Rolle namens AWSServiceRoleForKeyManagementServiceMultiRegionKeys , um multiregionale Schlüssel zu unterstützen. Diese serviceverknüpfte Rolle gibt AWS KMS die Berechtigung zum Synchronisieren von Änderungen am Schlüsselmaterial eines multiregionalen Primärschlüssels mit seinen Replikaten. AWS KMS erstellt diese Rolle nur, wenn Sie einen multiregionalen Primärschlüssel erstellen. Sie können diese serviceverknüpfte Rolle nicht direkt erstellen.

Die serviceverknüpfte Rolle AWSServiceRoleForKeyManagementServiceMultiRegionKeys vertraut darauf, dass `mrk.kms.amazonaws.com` die Rolle annimmt. Dies hat zur Folge, dass nur AWS KMS diese serviceverknüpfte Rolle annehmen kann. Die Berechtigungen in der Rolle werden auf die Aktionen beschränkt, die AWS KMSausführt, um das Schlüsselmaterial in verwandten multiregionalen Schlüsseln synchronisiert zu halten. AWS KMS werden keine weiteren Berechtigungen erteilt.

Weitere Informationen zur AWSServiceRoleForKeyManagementServiceMultiRegionKeys-Rolle, einschließlich einer Liste der Berechtigungen und Anweisungen zum Anzeigen der Rolle, Bearbeiten

der Rollenbeschreibung, Löschen der Rolle und Neuerstellen mithilfe von AWS KMS finden Sie unter Autorisieren von AWS KMS zum Synchronisieren der multiregionalen Schlüsseln.

AWS KMS-Aktualisierungen für AWS verwaltete Richtlinien

Anzeigen von Details zu Aktualisierungen für AWS-verwaltete Richtlinien für AWS KMS, seit dieser Dienst mit der Verfolgung dieser Änderungen begonnen hat. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der AWS KMS Dokumentverlauf-Seite.

Änderung	Beschreibung	Datum
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	AWS KMS hat die `ec2:DescribeNetworkInterfaces` Berechtigungen `ec2:DescribeVpcs` , und hinzugefügt`ec2:DescribeNetworkAcls` , um Änderungen in der VPC zu überwachen, die Ihren AWS CloudHSM Cluster enthält, sodass bei Ausfällen klare Fehlermeldungen bereitstellen AWS KMS kann.	10. November 2023
AWS KMS hat die Änderungsverfolgung gestartet	AWS KMS hat mit der Verfolgung von Änderungen für seine AWS-verwalteten Richtlinien begonnen.	10. November 2023

Verwenden von Hybrid-Post-Quantum-TLS mit AWS KMS

AWS Key Management Service (AWS KMS) unterstützt jetzt eine Hybrid-Post-Quantum-Schlüsselaustauschoption für das Transport-Layer-Security-Netzwerk-Verschlüsselungsprotokoll (TLS). Sie können diese TLS-Option verwenden, wenn Sie eine Verbindung zu AWS KMS-API-Endpunkten herstellen. Wir bieten dieses Feature an, bevor Post-Quantenalgorithmen standardisiert sind, damit Sie beginnen können, die Auswirkungen dieser Schlüsselaustauschprotokolle auf AWS

KMS Anrufe zu testen. Diese optionalen Hybrid-Post-Quantum-Schlüsselaustauschfunktionen sind mindestens so sicher wie die heute verwendete TLS-Verschlüsselung und bieten wahrscheinlich zusätzliche langfristige Sicherheitsvorteile. Sie wirken sich jedoch auf Latenz und Durchsatz im Vergleich zu den klassischen Schlüsselaustauschprotokollen aus, die heute verwendet werden.

Die Daten, die Sie an AWS Key Management Service (AWS KMS) senden, werden während der Übertragung durch die Verschlüsselung geschützt, die von einer TLS-Verbindung (Transport Layer Security) bereitgestellt wird. Die klassischen Cipher-Suites, die TLS-Sitzungen AWS KMS unterstützen, machen Brute-Force-Angriffe auf die Schlüsselaustauschmechanismen mit der aktuellen Technologie undurchführbar. Wenn jedoch in Zukunft das große Quantencomputing praktikabel wird, werden die klassischen Cipher-Suites, die in TLS-Schlüsselaustauschmechanismen verwendet werden, für diese Angriffe anfällig sein. Wenn Sie Anwendungen entwickeln, die auf die langfristige Vertraulichkeit von Daten beruhen, die über eine TLS-Verbindung übertragen werden, sollten Sie einen Plan zur Migration zur Postquantenkryptographie in Betracht ziehen, bevor große Quantencomputer zur Verwendung verfügbar werden. AWS arbeitet daran, sich auf diese Zukunft vorzubereiten, und wir möchten, dass Sie auch gut vorbereitet sind.

Um Daten zu schützen, die heute vor potenziellen zukünftigen Angriffen verschlüsselt werden, beteiligt AWS sich mit der kryptografischen Gemeinschaft an der Entwicklung von quantenresistenten oder Post-Quantum Algorithmen. Wir haben hybride Post-Quantum-Schlüsselaustausch-Cipher-Suites in AWS KMS implementiert, die klassische und Post-Quantum-Elemente kombinieren, um sicherzustellen, dass Ihre TLS-Verbindung mindestens so stark ist wie mit klassischen Cipher-Suites.

Diese hybriden Cipher-Suites sind in den meisten AWS-Regionen für Ihre Operations-Workloads verfügbar. Da sich die Leistungseigenschaften und die Bandbreitenanforderungen von Hybrid-Cipher-Suites jedoch von denen klassischer Schlüsselaustauschmechanismen unterscheiden, empfehlen wir, diese bei Ihren AWS KMS API-Aufrufen unter unterschiedlichen Bedingungen zu testen .

Feedback

Wie immer freuen wir uns über Ihr Feedback und Ihre Teilnahme an unseren Open Source-Repositories. Wir möchten besonders erfahren, wie Ihre Infrastruktur mit dieser neuen Variante des TLS-Datenverkehrs interagiert.

- Um Feedback zu diesem Thema zu geben, verwenden Sie den Link Feedback in der oberen rechten Ecke dieser Seite.
- Wir entwickeln diese Hybrid-Cipher-Suites in Open Source im s2n-tlsRepository auf GitHub. Um Feedback zur Benutzerfreundlichkeit der Cipher-Suites zu geben oder neue Testbedingungen oder Ergebnisse zu teilen, erstellen Sie eine Anforderung im s2n-tls-Repository.

- Wir schreiben Codebeispiele für die Verwendung von hybridem Post-Quantum-TLS mit AWS KMS im aws-kms-pq-tls-example GitHubRepository. Um Fragen zu stellen oder Ideen zur Konfiguration des HTTP-Clients oder AWS KMS -Clients für die Verwendung der Hybrid-Cipher-Suites zu teilen, erstellen Sie eine Anforderung im aws-kms-pq-tls-example Repository.

Unterstützte AWS-Regionen

Post-Quantum-TLS für AWS KMS ist in allen AWS-Regionen verfügbar, die AWS KMS unterstützt, mit Ausnahme von China (Peking) und China (Ningxia).

> **Note**
>
> AWS KMS unterstützt kein hybrides Post-Quantum-TLS für FIPS-Endpunkte in AWS GovCloud (US).

Eine Liste aller AWS KMS-Endpunkte für jede AWS-Region finden Sie unter AWS Key Management Service-Endpunkte und Kontingente im Allgemeine Amazon Web Services-Referenz. Weitere Informationen über FIPS-Endpunkte finden Sie unter FIPS-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

Über den Hybrid-Post-Quantum-Schlüsselaustausch in TLS

AWS KMS unterstützt Hybrid-Post-Quantum-Schlüsselaustausch-Cipher-Suites. Sie können die AWS SDK for Java 2.x und AWS Common Runtime (CRT) auf Linux-Systemen verwenden, um einen HTTP-Client für die Verwendung dieser Cipher-Suites zu konfigurieren. Wenn Sie dann mit Ihrem HTTP-Client eine Verbindung zu einem AWS KMS-Endpunkt herstellen, werden die Hybrid-Cipher-Suites verwendet.

Dieser HTTP-Client verwendet s2n-tls, eine Open-Source-Implementierung des TLS-Protokolls. Die Hybrid-Cipher-Suites , die s2n-tls verwendet, sind nur für den Schlüsselaustausch implementiert, nicht für die direkte Datenverschlüsselung. Während des Schlüsselaustauschs berechnen Client und Server den Schlüssel, den sie verwenden, um die Daten auf der Leitung zu verschlüsseln und zu entschlüsseln.

Die von s2n-tls verwendeten Algorithmen sind ein Hybrid, der Elliptic Curve Diffie-Hellman (ECDH), einen klassischen Schlüsselaustauschalgorithmus, der heute in TLS verwendet wird, mit Kyber kombiniert, einem Verschlüsselungs- und Schlüsselaustausch-Algorithmus für öffentliche Schlüssel,

den das National Institute for Standards and Technology (NIST) als seinen ersten Standard-Post-Quanten-Schlüsselaustausch–Algorithmus bezeichnet hat. Dieser Hybrid verwendet jeden der Algorithmen unabhängig, um einen Schlüssel zu generieren. Dann kombiniert es die beiden Schlüssel kryptografisch. Mit s2n-tls können Sie einen HTTP-Client so konfigurieren, dass er Post-Quantum-TLS bevorzugt, wodurch ECDH mit Kyber an erster Stelle in der Präferenzliste steht. Klassische Schlüsselaustauschalgorithmen sind in der Einstellungsliste enthalten, um die Kompatibilität sicherzustellen, aber sie sind in der Präferenzreihenfolge niedriger.

Wenn die laufende Forschung ergibt, dass dem Kyber-Algorithmus die erwartete Post-Quantum-Stärke fehlt, ist der Hybrid-Schlüssel immer noch mindestens so stark wie der einzelne ECDH-Schlüssel, der derzeit verwendet wird. Bis die Forschung zu Post-Quanten-Algorithmen abgeschlossen ist, empfehlen wir, Hybrid-Algorithmen zu verwenden, anstatt nur Post-Quanten-Algorithmen zu verwenden.

Verwenden von Hybrid-Post-Quantum-TLS mit AWS KMS

Sie können Hybrid-Post-Quantum-TLS für Ihre Anrufe an verwenden AWS KMS. Beachten Sie beim Einrichten der HTTP-Client-Testumgebung die folgenden Informationen:

Verschlüsselung während der Übertragung

Die Hybrid-Cipher-Suites in s2n-tls werden nur für die Verschlüsselung während der Übertragung verwendet. Sie schützen Ihre Daten, während sie von Ihrem Client zum AWS KMS-Endpunkt unterwegs sind. AWS KMS verwendet diese Cipher-Suites nicht, um Daten unter AWS KMS keys zu verschlüsseln.

Wenn AWS KMS stattdessen Ihre Daten unter KMS-Schlüssel verschlüsselt, verwendet es symmetrische Kryptographie mit 256-Bit-Schlüsseln und den Advanced-Encryption-Standard im Galois-Counter-Mode-Algorithmus (AES-GCM), der bereits quantenresistent ist. Theoretische zukünftige, groß angelegte Quantenrechnungsangriffe auf Verschlüsselungstexte, die unter 256-Bit-AES-GCM-Schlüsseln erstellt wurden, reduzieren die effektive Sicherheit des Schlüssels auf 128-Bits. Diese Sicherheitsstufe reicht aus, um Brute-Force-Angriffe auf AWS KMS-Verschlüsselungstexte undurchführbar zu machen.

Unterstützte Systeme

Die Verwendung der Hybrid-Cipher-Suites in s2n-tls wird derzeit nur auf Linux-Systemen unterstützt. Darüber hinaus werden diese Cipher-Suites nur in SDKs unterstützt, die die AWS-Common-Runtime unterstützen, z. B. AWS SDK for Java 2.x Ein Beispiel finden Sie unter Konfigurieren von Hybrid-Post-Quantum-TLS.

AWS KMSEndpunkte

Verwenden Sie bei Verwendung der Hybrid-Cipher-Suites den AWS KMS Standardendpunkt. Die Hybrid-Cipher-Suites in s2n-tls sind nicht kompatibel mit den FIPS 140-2 validierten Endpunkten für AWS KMS.

Wenn Sie einen HTTP-Client so konfigurieren, dass er Post-Quantum-TLS-Verbindungen mit s2n-tls bevorzugt, stehen die Post-Quantum-Ciphers in der Liste der bevorzugten Ciphers an erster Stelle. Die Voreinstellungsliste enthält jedoch die klassischen, nicht-hybriden Verschlüsselungen, die in der Präferenzreihenfolge aus Gründen der Kompatibilität niedriger sind. Wenn Sie einen HTTP-Client konfigurieren möchten, der Post-Quantum-Verschlüsselungspräferenz mit einem von AWS KMS FIPS 140-2 validierten Endpunkt verwendet, verhandelt s2n-tls eine klassische, nicht-hybride Schlüsselaustausch-Cipher.

Eine Liste aller AWS KMS-Endpunkte für jede AWS-Region finden Sie unter AWS Key Management Service-Endpunkte und Kontingente im Allgemeine Amazon Web Services-Referenz. Weitere Informationen über FIPS-Endpunkte finden Sie unter FIPS-Endpunkte in der Allgemeine Amazon Web Services-Referenz.

Erwartete Leistung

Unsere frühen Benchmark-Tests zeigen, dass die Hybrid-Cipher-Suites in s2n-tls langsamer sind als klassische TLS-Cipher-Suites. Der Effekt variiert je nach Netzwerkprofil, CPU-Geschwindigkeit, Anzahl der Kerne und Aufrufrate. Leistungstestergebnisse finden Sie unter So stimmen Sie TLS für hybride Post-Quanten-Kryptografie mit Kyber ab.

Konfigurieren von Hybrid-Post-Quantum-TLS

In diesem Verfahren fügen Sie eine Maven-Abhängigkeit für den AWS Common Runtime HTTP Client hinzu. Anschließend konfigurieren Sie einen HTTP-Client, der Post-Quantum-TLS bevorzugt. Erstellen Sie dann einen AWS KMS-Client, der den HTTP-Client verwendet.

Eine vollständige Liste von Arbeitsbeispielen zum Konfigurieren und Verwenden von hybriden Post-Quantum-TLS mit AWS KMS, finden Sie im aws-kms-pq-tls-example-Repository.

> **Note**
>
> Der AWS Common Runtime HTTP Client, der als Vorschau verfügbar war, wurde im Februar 2023 allgemein verfügbar. In dieser Version werden die `tlsCipherPreference-`

Klasse und der `tlsCipherPreference()`-Methodenparameter durch den `postQuantumTlsEnabled()`-Methodenparameter ersetzt. Wenn Sie dieses Beispiel in der Vorversion verwendet haben, müssen Sie Ihren Code aktualisieren.

1. Fügen Sie den AWS-Common-Runtime-Client zu Ihren Maven-Abhängigkeiten hinzu. Wir empfehlen, die neueste verfügbare Version zu verwenden.

   Diese Anweisung fügt beispielsweise die Version `2.20.0` des AWS Common Runtime Clients zu Ihren Maven-Abhängigkeiten hinzu.

   ```
   <dependency>
       <groupId>software.amazon.awssdk</groupId>
       <artifactId>aws-crt-client</artifactId>
       <version>2.20.0</version>
   </dependency>
   ```

2. Um die Hybrid-Post-Quantum-Cipher-Suites zu aktivieren, fügen Sie das AWS SDK for Java 2.x zu Ihrem Projekt hinzu und initialisieren Sie es. Aktivieren Sie dann die hybriden Post-Quantum-Cipher-Suites auf Ihrem HTTP-Client, wie im folgenden Beispiel gezeigt.

   Dieser Code verwendet den `postQuantumTlsEnabled()`-Methodenparameter, um einen AWS-Common-Runtime-HTTP-Client zu konfigurieren, der die empfohlene hybride Post-Quantum-Cipher Suite ECDH mit Kyber bevorzugt. Dann verwendet er den konfigurierten HTTP-Client, um eine Instance des AWS KMS-asynchronen Clients zu erstellen, `KmsAsyncClient`. Nachdem dieser Code abgeschlossen ist, verwenden alle AWS KMS-API-Anfragen auf der `KmsAsyncClient`-Instance hybrides Post-Quantum-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. Testen Sie Ihre AWS KMS-Aufrufe mit hybridem Post-Quantum-TLS.

Wenn Sie AWS KMS API-Vorgänge auf dem konfigurierten AWS KMS Client aufrufen, werden Ihre Aufrufe mithilfe von hybriden Post-Quantum-TLS an den AWS KMS Endpunkt übertragen. Um Ihre Konfiguration zu testen, führen Sie einen AWS KMS-API-Aufruf aus, z. B. `ListKeys`.

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

Testen von Hybrid-Post-Quantum-TLS mit AWS KMS

Ziehen Sie in Betracht, die folgenden Tests mit Hybrid-Cipher-Suites für Ihre Anwendungen auszuführen, die AWS KMS aufrufen.

- Führen Sie Belastungstests und Benchmarks aus. Die Hybrid-Cipher-Suites funktionieren anders als herkömmliche Schlüsselaustauschalgorithmen. Möglicherweise müssen Sie Ihre Verbindungszeitüberschreitungen anpassen, um längere Handshake-Zeiten zu ermöglichen. Wenn Sie innerhalb einer AWS Lambda Funktion ausgeführt werden, verlängern Sie die Einstellung für das Ausführungszeitlimit.
- Versuchen Sie, eine Verbindung von verschiedenen Standorten herzustellen. Abhängig vom Netzwerkpfad Ihrer Anforderung können Sie feststellen, dass Zwischenhosts, Proxys oder Firewalls mit Deep Packet Inspection (DPI) die Anforderung blockieren. Dies kann auf die Verwendung der neuen Cipher Suites im ClientHello Teil des TLS-Handshakes oder auf die größeren Schlüsselaustauschnachrichten zurückzuführen sein. Wenn Sie Probleme bei der Behebung dieser Probleme haben, arbeiten Sie mit Ihrem Sicherheitsteam oder IT-Administratoren zusammen, um die entsprechende Konfiguration zu aktualisieren und die Blockierung der neuen TLS-Cipher-Suites aufzuheben.

Weitere Informationen zu Post-Quantum-TLS finden Sie unter AWS KMS

Weitere Informationen zur Verwendung von Hybrid-Post-Quantum-TLS in AWS KMS, finden Sie unter den folgenden Ressourcen.

- Weitere Informationen zur Post-Quantum-Kryptographie bei AWS finden Sie unter Post-Quantum-Kryptografie, einschließlich Links zu Blogbeiträgen und Forschungsarbeiten.
- Weitere Informationen zu s2n-tls finden Sie unter Introducing s2n-tls, a New Open Source TLS Implementation und Using s2n-tls.

- Informationen zum AWS Common Runtime HTTP Client finden Sie unter Konfigurieren des AWS-CRT-basierten HTTP-Clients im AWS SDK for Java 2.x-Entwicklerhandbuch .
- Weitere Informationen zum Post-Quantum-Kryptographie-Projekt am National Institute for Standards and Technology (NIST) finden Sie unter Post-Quantum-Kryptographie.
- Informationen zur Standardisierung der Post-Quantum-Kryptografie durch NIST finden Sie unter Standardisierung der Post-Quantum-Kryptografie.

Bestimmen des Zugriffs auf AWS KMS keys

Um vollständig zu verstehen, wer oder was derzeit Zugriff auf einen AWS KMS key hat, müssen Sie die Schlüsselrichtlinie des KMS-Schlüssels, alle für den KMS-Schlüssel geltenden Erteilungen und möglicherweise alle AWS Identity and Access Management (IAM)-Richtlinien prüfen. Sie können damit den Umfang der potenziellen Nutzung eines KMS-Schlüssels bestimmen, oder Compliance- oder Auditing-Anforderungen erfüllen. Die folgenden Themen können Ihnen dabei helfen, eine vollständige Liste der AWS-Prinzipale (Identitäten) zu erzeugen, die derzeit Zugriff auf einen KMS-Schlüssel haben.

Themen

Untersuchen der Schlüsselrichtlinie

Schlüsselrichtlinien sind die primäre Methode zur Zugriffssteuerung für KMS-Schlüssel. Jeder KMS-Schlüssel besitzt genau eine Schlüsselrichtlinie.

Wenn eine Schlüsselrichtlinie aus der Standard-Schlüsselrichtlinie besteht oder sie enthält, berechtigt die Schlüsselrichtlinie IAM-Administratoren im Konto dazu, mithilfe von IAM-Richtlinien den Zugriff auf den KMS-Schlüssel zu steuern. Wenn die Schlüsselrichtlinie einem anderen AWS-Konto die Berechtigung zur Verwendung des KMS-Schlüssels erteilt, können die IAM-Administratoren im externen Konto diese Berechtigungen anhand von IAM-Richtlinien delegieren. Um die komplette Liste der Prinzipale mit Zugriff auf den KMS-Schlüssel zu bestimmen, untersuchen Sie die IAM-Richtlinien.

Um die Schlüsselrichtlinie eines vom AWS KMS Kunden verwalteten Schlüssels oder Von AWS verwalteter Schlüssel in Ihrem Konto anzuzeigen, verwenden Sie die - AWS Management Console oder die -GetKeyPolicyOperation in der AWS KMS-API. Um die Schlüsselrichtlinie anzeigen zu können, müssen Sie über `kms:GetKeyPolicy`-Berechtigungen für den KMS-Schlüssel verfügen. Anweisungen zum Anzeigen der Schlüsselrichtlinie für einen KMS-Schlüssel finden Sie unter the section called “Anzeigen einer Schlüsselrichtlinie”.

Überprüfen Sie das Schlüssel-Richtliniendokument und notieren Sie alle Hauptpunkte, die in jedem `Principal` Element der Richtlinie aufgeführt ist. In einer Richtlinienanweisung mit der Auswirkung `Allow` haben die IAM-Benutzer, IAM-Rollen und AWS-Konten in dem Element `Principal` Zugriff auf diesen KMS-Schlüssel.

> **Note**
>
> Setzen Sie den Prinzipal nicht auf ein Sternchen (*) in einer Schlüsselrichtlinienanweisung, die Berechtigungen erlaubt, es sei denn, Sie verwenden Bedingungen, um die Schlüsselrichtlinie einzuschränken. Ein Sternchen gibt jeder Identität in jedem AWS-Konto die Berechtigung, den KMS-Schlüssel zu verwenden, es sei denn, eine andere Richtlinienanweisung verweigert dies explizit. Benutzer in anderen AWS-Konten können Ihren KMS-Schlüssel verwenden, wenn sie in ihrem eigenen Konto über entsprechende Berechtigungen verfügen.

Die folgenden Beispiele verwenden die Richtlinien-Anweisungen in der Standard-Schlüsselrichtlinie, um dies zu demonstrieren.

Example Richtlinienanweisung 1

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

In der Richtlinienanweisung 1 ist `arn:aws:iam::111122223333:root` ein AWS-Konto-Prinzipal, der sich auf das AWS-Konto 111122223333 bezieht. (Er ist nicht der Root-Benutzer des Kontos.) Standardmäßig ist eine Richtlinienanweisung wie diese im Schlüssel-richtliniendokument vorhanden,

wenn Sie mit der AWS Management Console einen neuen KMS-Schlüssel erstellen, und Sie einen neuen KMS-Schlüssel programmgesteuert erstellen, aber keine Schlüsselrichtlinie angeben.

Ein Schlüsselrichtliniendokument mit einer Anweisung, die den Zugriff auf das AWS-Konto erlaubt, aktiviert IAM-Richtlinien im Konto, um den Zugriff auf den KMS-Schlüssel zu erlauben. Das bedeutet, dass Benutzer und Rollen im Konto Zugriff auf den KMS-Schlüssel haben könnten, auch wenn sie explizit nicht als Prinzipale im Schlüsselrichtliniendokument aufgelistet sind. Untersuchen Sie alle IAM-Richtlinien in allen AWS-Konten, die als Prinzipale aufgeführt sind, um zu ermitteln, ob sie Zugriff auf diesen KMS-Schlüssel erlauben.

Example Richtlinienanweisung 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": "*"
}
```

In Richtlinienanweisung 2 `arn:aws:iam::111122223333:role/KMSKeyAdmins` bezieht sich auf die IAM-Rolle mit dem Namen KMS KeyAdmins im AWS-Konto 111122223333. Benutzer, die diese Rolle annehmen dürfen, dürfen die in der Richtlinienanweisung aufgeführten Aktionen ausführen, die die administrativen Aktionen für die Verwaltung eines KMS-Schlüssels sind.

Example Richtlinienanweisung 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": "*"
}
```

In Richtlinienanweisung 3 `arn:aws:iam::111122223333:role/EncryptionApp` bezieht sich auf die IAM-Rolle mit dem Namen EncryptionApp in AWS-Konto 111122223333. Prinzipale, die diese Rolle annehmen dürfen, dürfen die in der Richtlinienanweisung aufgeführten Aktionen durchführen, die die kryptografischen Operationen für einen symmetrischen KMS-Schlüssel enthalten.

Example Richtlinienanweisung 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}}
}
```

In Richtlinienanweisung 4 `arn:aws:iam::111122223333:role/EncryptionApp` bezieht sich auf die IAM-Rolle mit dem Namen EncryptionApp in AWS-Konto 111122223333. Prinzipal, die diese Rolle annehmen dürfen, dürfen die in der Richtlinienanweisung aufgeführten Aktionen ausführen. Diese Aktionen in Kombination mit den erlaubten Aktionen in der Beispiel-Richtlinienanweisung 3 sind diejenigen, die zum Delegieren der Verwendung des KMS-Schlüssels für die meisten AWS-Services, die mit AWS KMS integriert sind, erforderlich sind, insbesondere für die Services, die Erteilungen verwenden. Der kms:GrantIsForAWSResource-Wert im -`Condition`Element stellt sicher, dass die Delegierung nur zulässig ist, wenn der Delegierte ein -AWSService ist, der in integriert ist AWS KMS und Erteilungen für die Autorisierung verwendet.

Die unterschiedlichen Möglichkeiten zum Angeben eines Prinzipals in einem Schlüsselrichtlinien-Dokument finden Sie unter Angeben eines Prinzipals im IAM-Benutzerhandbuch.

Weitere Informationen zu AWS KMS-Schlüsselrichtlinien finden Sie unter Wichtige Richtlinien in AWS KMS.

Untersuchen von IAM-Richtlinien

Zusätzlich zu den Schlüsselrichtlinien und Berechtigungen können Sie auch IAM-Richtlinien verwenden, um den Zugriff auf einen KMS-Schlüssel zu erlauben. Weitere Informationen darüber, wie IAM-Richtlinien und Schlüsselrichtlinien zusammen funktionieren, finden Sie unter Fehlerbehebung beim Schlüsselzugriff.

Um zu bestimmen, welche Prinzipale derzeit über IAM-Richtlinien auf einen KMS-Schlüssel zugreifen können, verwenden Sie das browserbasierte IAM-Richtliniensimulator-Tool. Alternativ können Sie Anforderungen an die IAM-API durchführen.

Untersuchen von IAM-Richtlinien

- Untersuchen von IAM-Richtlinien mit dem IAM-Richtliniensimulator
- Untersuchen von IAM-Richtlinien mit der IAM-API

Untersuchen von IAM-Richtlinien mit dem IAM-Richtliniensimulator

Mit dem IAM-Richtliniensimulator können Sie erfahren, welche Prinzipale über eine IAM-Richtlinie Zugriff auf einen KMS-Schlüssel haben.

So verwenden Sie den IAM-Richtliniensimulator, um den Zugriff auf einen KMS-Schlüssel zu bestimmen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie den IAM-Richtliniensimulator unter https://policysim.aws.amazon.com/.
2. Wählen Sie im Feld Benutzer, Gruppen und Rollen den Benutzer, Gruppe oder Rolle, deren Richtlinien Sie simulieren möchten.
3. (Optional) Deaktivieren Sie das Kontrollkästchen neben den Richtlinien, die Sie von der Simulation auslassen möchten. Um alle Richtlinien zu simulieren, markieren Sie alle Richtlinien.
4. Führen Sie im Bereich Richtliniensimulator die folgenden Schritte aus:

    a. Wählen Sie für Service wählen die Option Key Management Service.

   b. Um spezifische AWS KMS-Aktionen für die Select actions (Aktionen auswählen) zu simulieren, wählen Sie die zu simulierende Aktionen aus. Um alle AWS KMS-Aktionen zu simulieren, wählen Sie Select All (Alle auswählen) aus.

5. (Optional) Der Richtliniensimulator simuliert standardmäßig einen Zugriff auf alle KMS-Schlüssel. Um den Zugriff auf einen bestimmten KMS-Schlüssel zu simulieren, wählen Sie Simulation Settings (Simulationseinstellungen) und geben Sie dann den Amazon-Ressourcennamen (ARN) des zu simulierenden KMS-Schlüssels ein.
6. Wählen Sie Run Simulation (Simulation ausführen).

Sie können die Ergebnisse der Simulation im Abschnitt Ergebnisse sehen. Wiederholen Sie die Schritte 2 bis 6 für alle Benutzer, Gruppen und Rollen im AWS-Konto.

Untersuchen von IAM-Richtlinien mit der IAM-API

Sie können mit der IAM-API programmgesteuert IAM-Richtlinien untersuchen. Die folgenden Schritte bieten eine allgemeine Übersicht darüber:

1. Verwenden Sie für jede , die als Prinzipal in der Schlüsselrichtlinie AWS-Konto aufgeführt ist (d. h. für jeden AWS Kontoprinzipal, der in diesem Format angegeben ist: `"Principal": {"AWS": "arn:aws:iam::111122223333:root"}`), die - ListUsers und -ListRolesOperationen in der IAM-API, um alle Benutzer und Rollen im Konto abzurufen.
2. Verwenden Sie für jeden Benutzer und jede Rolle in der Liste die -SimulatePrincipalPolicyOperation in der IAM-API, wobei Sie die folgenden Parameter übergeben:
   - Geben Sie für `PolicySourceArn` den Amazon-Ressourcennamen (ARN) für einen Benutzer oder eine Rolle aus der Liste an. Sie können nur jeweils einen `PolicySourceArn` pro `SimulatePrincipalPolicy`-Anforderung angeben. Deshalb müssen Sie diese Operation mehrfach aufrufen – jeweils einmal für jeden Benutzer und jede Rolle in der Liste.
   - Geben Sie für die Liste `ActionNames` alle zu simulierende AWS KMS-API-Aktionen an. Um alle AWS KMS-API-Aktionen zu simulieren, verwenden Sie `kms:*`. Um einzelne AWS KMS-API-Aktionen zu testen, stellen Sie jeder API-Aktion "`kms:`" voran, z. B. "`kms:ListKeys`". Eine vollständige Liste aller AWS KMS-API-Aktionen finden Sie unter Aktionen in der AWS Key Management Service-API-Referenz.
   - (Optional) Um zu bestimmen, ob die Benutzer oder Rollen Zugriff auf bestimmte KMS-Schlüssel haben, verwenden Sie den `ResourceArns`-Parameter, um eine Liste der Amazon-Ressourcennamen (ARNs) der KMS-Schlüssel anzugeben. Vermeiden Sie den

`ResourceArns`-Parameter, um zu prüfen, ob die Benutzer oder Rollen Zugriff auf irgendwelche KMS-Schlüssel haben.

IAM antwortet auf jede `SimulatePrincipalPolicy`-Anforderung mit einer Bewertungsentscheidung: `allowed`, `explicitDeny`, oder `implicitDeny`. Für jede Antwort mit der Bewertung `allowed` enthält die Antwort den Namen der spezifischen AWS KMS-API-Produktion, die zulässig ist. Darüber hinaus enthält sie den ARN des KMS-Schlüssels, der in der Bewertung verwendet wurde, falls vorhanden.

Prüfen von Erteilungen

Erteilungen sind erweiterte Mechanismen zur Definition von Berechtigungen, die Sie oder ein mit AWS KMS integrierter AWS-Service verwenden können, um festzulegen, wie und wann ein KMS-Schlüssel verwendet werden darf. Erteilungen werden einem KMS-Schlüssel angefügt. Jede Erteilung enthält den Prinzipal, dem die Berechtigung zur Verwendung des KMS-Schlüssels erteilt wird, sowie eine Liste der erlaubten Produktionen. Berechtigungen sind eine Alternative zu den Schlüsselrichtlinien und eignen sich für bestimmte Anwendungsfälle. Weitere Informationen finden Sie unter Erteilungen in AWS KMS.

Um eine Liste der Erteilungen für einen KMS-Schlüssel abzurufen, verwenden Sie die -AWS KMSListGrantsOperation. Sie können die Erteilungen für einen KMS-Schlüssel untersuchen, um zu bestimmen, wer oder was derzeit über diese Erteilungen Zugriff auf den KMS-Schlüssel hat. Das folgende Beispiel ist eine JSON-Darstellung einer Erteilung, die vom Befehl list-grants in der AWS CLI abgerufen wurde.

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

Um herauszufinden, wer oder was Zugriff auf den KMS-Schlüssel hat, suchen Sie nach dem `"GranteePrincipal"`-Element. Im vorherigen Beispiel ist der Empfänger-Prinzipal ein Benutzer der angenommene Rolle, der mit der EC2-Instance i-5d476fab verknüpft ist. Die EC2-Infrastruktur verwendet diese Rolle, um das verschlüsselten EBS-Volume vol-5cccfb4e an die Instance anzufügen. In diesem Fall ist die EC2-Infrastruktur-Rolle berechtigt, den KMS-Schlüssel zu verwenden, da Sie zuvor ein verschlüsseltes EBS-Volume erstellt haben, das von diesem KMS-Schlüssel geschützt wird. Anschließend haben Sie das Volume an eine EC2-Instance angefügt.

Hier ein weiteres Beispiel einer JSON-Darstellung einer Erteilung, die vom Befehl list-grants in der AWS CLI abgerufen wurde. Im folgenden Beispiel ist der Empfänger-Prinzipal ein weiteres AWS-Konto.

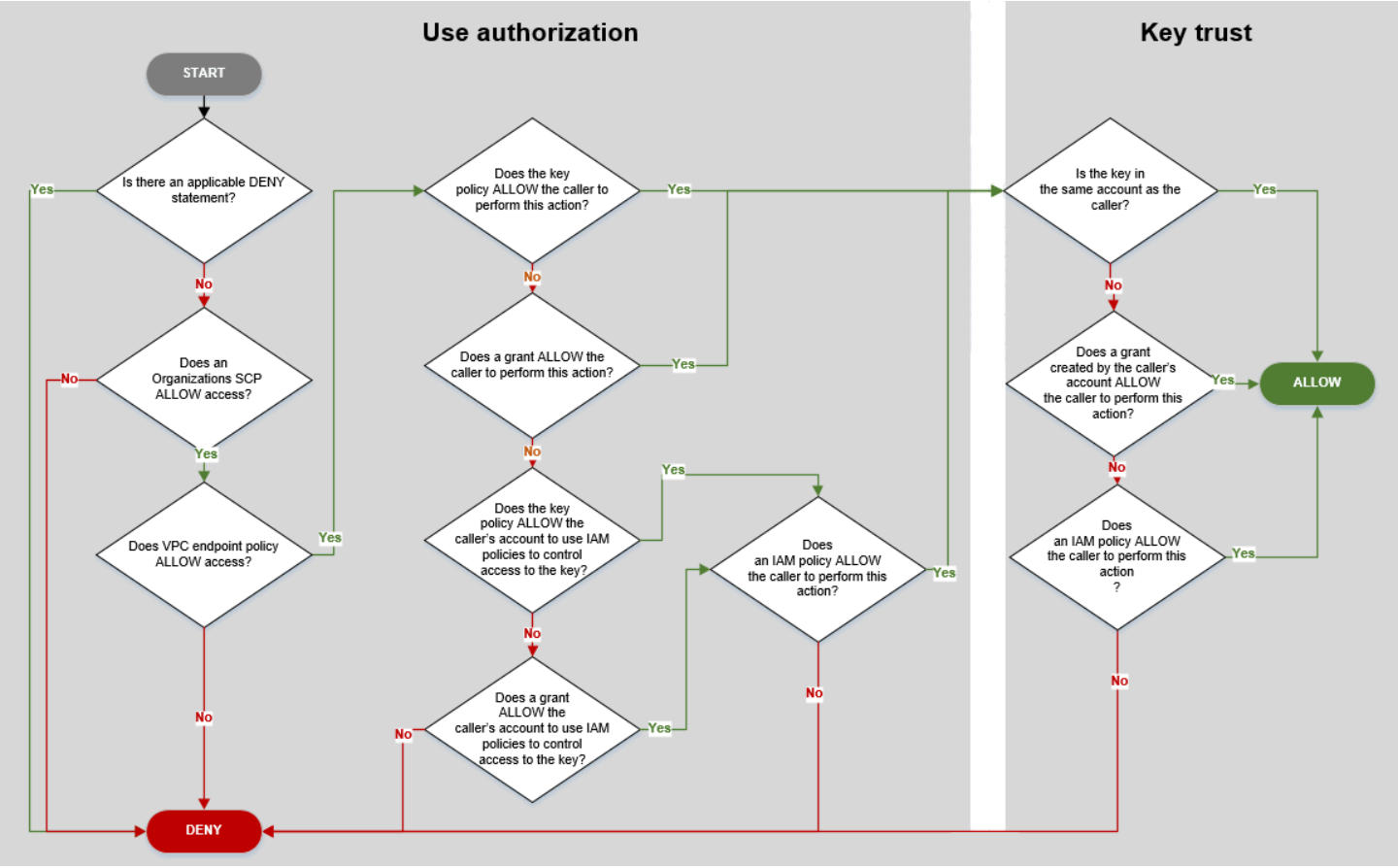
```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

Fehlerbehebung beim Schlüsselzugriff

Bei der Autorisierung des Zugriffs auf einen KMS-Schlüssel evaluiert AWS KMS Folgendes:

- Die Schlüsselrichtlinie, die dem KMS-Schlüssel angefügt ist. Die Schlüsselrichtlinie ist stets in dem AWS-Konto und der Region definiert, das/die den KMS-Schlüssel besitzt.
- Alle IAM-Richtlinien, die dem Benutzer oder der Rolle, die die Anforderung stellt, angefügt sind Die IAM-Richtlinien, die den Gebrauch eines KMS-Schlüssels durch einen Prinzipal regeln, sind immer in dem AWS-Konto des Prinzipals definiert.
- Alle Erteilungen für den KMS-Schlüssel.
- Andere Richtlinientypen, die möglicherweise für die Anforderung zur Verwendung des KMS-Schlüssels gelten, z. B. AWS Organizations-Service-Kontrollrichtlinien und VPC-Endpunktrichtlinien. Diese Richtlinien sind optional und erlauben standardmäßig alle Aktionen. Sie können sie jedoch verwenden, um Berechtigungen zu beschränken, die ansonsten Prinzipalen gewährt werden.

AWS KMS wertet die Richtlinien-Mechanismen zusammen aus, um zu bestimmen, ob der Zugriff auf den KMS-Schlüssel erlaubt oder abgelehnt werden soll. Hierzu verwendet AWS KMS einen Prozess ähnlich dem im folgenden Flussdiagramm dargestellten. Das folgende Flussdiagramm bietet eine visuelle Darstellung des Richtlinienauswertungsprozesses.



Dieses Flussdiagramm ist in zwei Teile unterteilt. Die Teile sind sequenziell angeordnet, werden in der Regel aber gleichzeitig ausgewertet.

- Use Authorization (Autorisierung verwenden) legt fest, ob Sie zur Verwendung eines KMS-Schlüssels, basierend auf dessen Schlüsselrichtlinie, IAM-Richtlinien und Erteilungen, berechtigt sind.
- Key trust (Schlüsselbasierte Vertrauensstellung) legt fest, ob Sie einem KMS-Schlüssel vertrauen sollten, den Sie verwenden dürfen. Im Allgemeinen vertrauen Sie den Ressourcen in Ihrem AWS-Konto. Sie können auch Vertrauen in die Nutzung von KMS-Schlüsseln in einem anderen AWS-Konto haben, wenn eine Erteilung oder eine IAM-Richtlinie in Ihrem Konto Ihnen die Verwendung des KMS-Schlüssels erlaubt.

Aus diesem Flussdiagramm geht hervor, warum einem Aufrufer die Berechtigung für die Verwendung eines KMS-Schlüssels gewährt oder verweigert wurde. Sie können sie auch zur Bewertung Ihrer Richtlinien und Genehmigungen heranziehen. Das Flussdiagramm veranschaulicht beispielsweise, dass einem Aufrufer der Zugriff durch eine explizite `DENY`-Anweisung oder durch das Weglassen einer expliziten `ALLOW`-Anweisung in der Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung verwehrt werden kann.

Das Flussdiagramm kann einige gängige Berechtigungsszenarien erläutern.

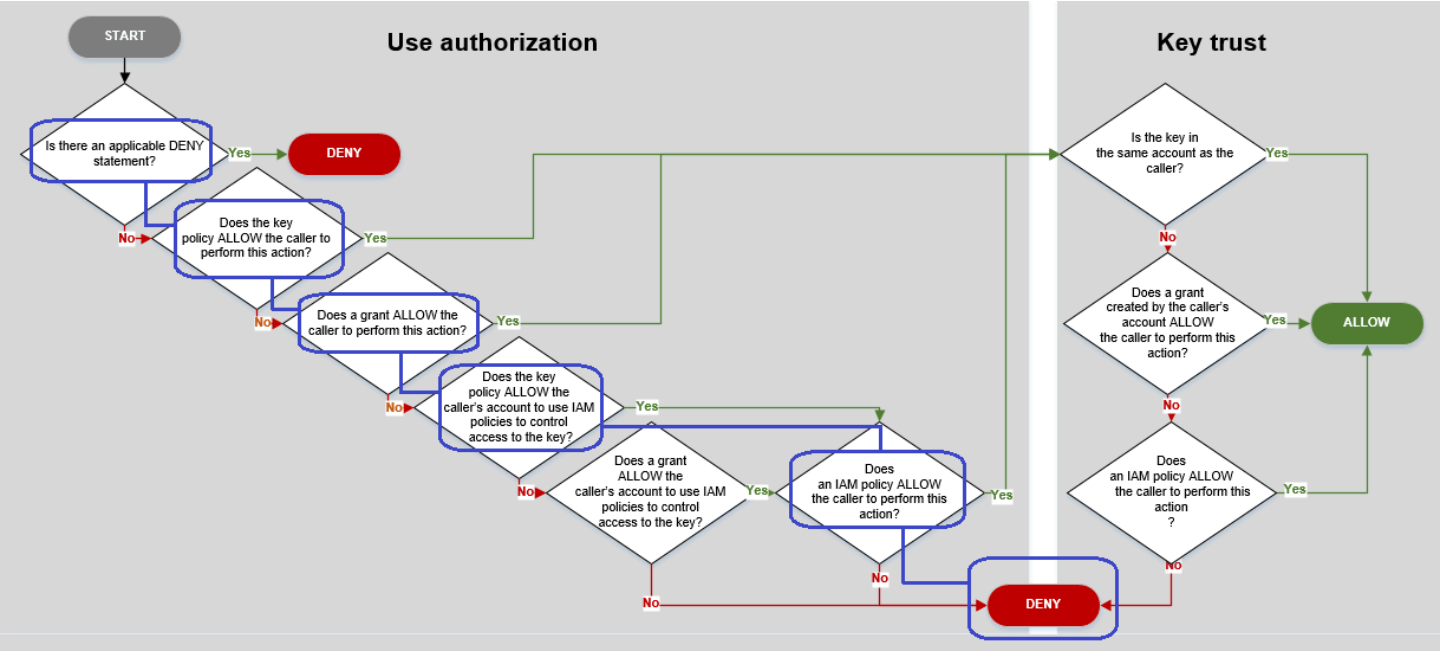
Beispiele für Berechtigungen

- Beispiel 1: Dem Benutzer wird der Zugriff auf einen KMS-Schlüssel in seinem AWS-Konto verweigert
- Beispiel 2: Der Benutzer übernimmt eine Rolle mit der Berechtigung zur Verwendung eines KMS-Schlüssels in einem anderen AWS-Konto.

Beispiel 1: Dem Benutzer wird der Zugriff auf einen KMS-Schlüssel in seinem AWS-Konto verweigert

Alice ist ein IAM-Benutzer im AWS-Konto 111122223333. Ihr wurde der Zugriff auf einen KMS-Schlüssel im selben AWS-Konto verweigert. Warum kann Alice den KMS-Schlüssel nicht verwenden?

Alice erhält in diesem Fall keinen Zugriff auf den KMS-Schlüssel, da keine Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung vorliegt, die ihr die erforderlichen Berechtigungen erteilen könnte. Die Schlüsselrichtlinie für KMS-Schlüssel erlaubt es dem AWS-Konto, IAM-Richtlinien zu verwenden, um den Zugriff auf den KMS-Schlüssel zu steuern. Keine IAM-Richtlinie erteilt Alice jedoch die Berechtigung zur Verwendung des KMS-Schlüssels.



Beachten Sie die relevanten Richtlinien für dieses Beispiel.

- Der KMS-Schlüssel, den Alice verwenden möchte, untersteht der Standardschlüsselrichtlinie. Diese Richtlinie erlaubt es dem AWS-Konto, das den KMS-Schlüssel besitzt, IAM-Richtlinien zum Steuern des Zugriffs auf den KMS-Schlüssel zu verwenden. Dieser Schlüssel erfüllt die im Abschnitt ERLAUBT die Schlüsselrichtlinie dem Aufruferkonto die Verwendung von IAM-Richtlinien zur Kontrolle des Zugriffs auf den Schlüssel? des Flussdiagramms beschriebene Bedingung.

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

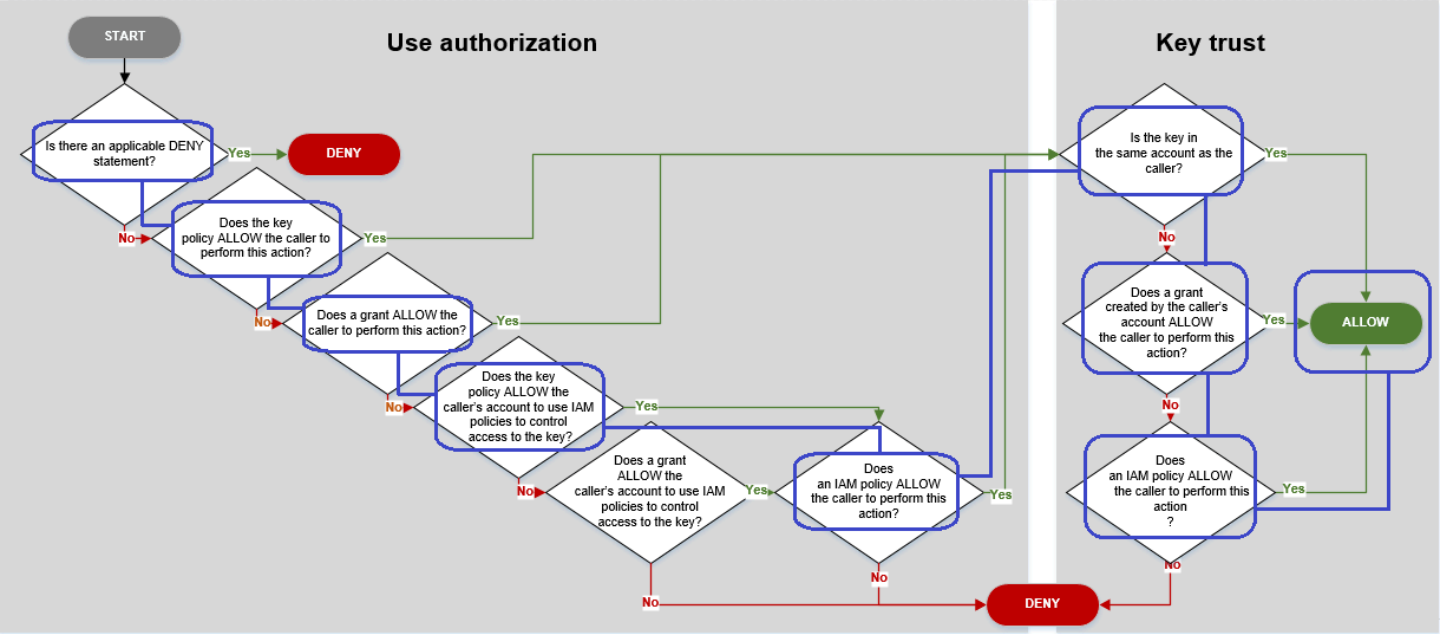
- Keine Schlüsselrichtlinie, IAM-Richtlinie oder Erteilung erteilt Alice jedoch die Berechtigung zur Verwendung des KMS-Schlüssels. Daher wird Alice die Berechtigung zur Verwendung des KMS-Schlüssels verweigert.

Beispiel 2: Der Benutzer übernimmt eine Rolle mit der Berechtigung zur Verwendung eines KMS-Schlüssels in einem anderen AWS-Konto.

Bob ist ein Benutzer in Konto 1 (111122223333). Er ist berechtigt, in Konto 2 (444455556666) einen KMS-Schlüssel in kryptografischen Produktionen zu verwenden. Wie ist das möglich?

> **Tip**
>
> Denken Sie beim Auswerten von kontoübergreifenden Berechtigungen daran, dass die Schlüsselrichtlinie im Konto des KMS-Schlüssels angegeben wird. Die IAM-Richtlinie wird im Konto des Aufrufers angegeben, auch wenn sich der Aufrufer in einem anderen Konto befindet. Weitere Informationen zum Bereitstellen des kontoübergreifenden Zugriffs auf KMS-Schlüssel finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

- Die Schlüsselrichtlinie für den KMS-Schlüssel in Konto 2 erlaubt Konto 2 die Verwendung von IAM-Richtlinien zum Steuern des Zugriffs auf den KMS-Schlüssel.
- Die Schlüsselrichtlinie für den KMS-Schlüssel in Konto 2 erlaubt Konto 1 die Verwendung des KMS-Schlüssels in kryptografischen Produktionen. Konto 1 muss jedoch seinen Prinzipalen anhand von IAM-Richtlinien Zugriff auf den KMS-Schlüssel gewähren.
- Eine IAM-Richtlinie in Konto 1 erlaubt es der `Engineering`-Rolle, den KMS-Schlüssel in Konto 2 für kryptografische Produktionen zu verwenden.
- Bob, ein Benutzer in Konto 1, verfügt über die Berechtigung, die `Engineering`-Rolle zu übernehmen.
- Bob kann diesem KMS-Schlüssel vertrauen, denn obwohl sich dieser nicht in seinem Konto befindet, ist es ihm über eine IAM-Richtlinie in seinem Konto explizit erlaubt, diesen KMS-Schlüssel zu verwenden.



Betrachten wir einmal die Richtlinien, die es Bob, einem Benutzer in Konto 1, ermöglichen, den KMS-Schlüssel in Konto 2 zu verwenden.

- Die Schlüsselrichtlinie für den KMS-Schlüssel erlaubt es Konto 2 (444455556666, das Konto, das den KMS-Schlüssel besitzt), den Zugriff auf den KMS-Schlüssel anhand von IAM-Richtlinien zu steuern. Diese Schlüsselrichtlinie erlaubt außerdem Konto 1 (111122223333) die Verwendung des KMS-Schlüssels in kryptografischen Produktionen (angegeben im `Action`-Element der Richtlinienanweisung). Niemand in Konto 1 kann jedoch den KMS-Schlüssel in Konto 2 verwenden, sofern in Konto 1 keine IAM-Richtlinien definiert sind, die den Prinzipalen Zugriff auf den KMS-Schlüssel gewähren.

  Im Flussdiagramm erfüllt diese Schlüsselrichtlinie in Konto 2 die Bedingung Does the key policy ALLOW the caller's account to use IAM policies to control access to the key? (ERLAUBT die Schlüsselrichtlinie dem Aufrufer-Konto die Verwendung von IAM-Richtlinien zur Steuerung des Zugriffs auf den Schlüssel?)

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

- Eine IAM-Richtlinie im AWS-Konto des Aufrufers (Konto 1, 111122223333) erteilt dem Prinzipal die Berechtigung, kryptografische Produktionen mithilfe des KMS-Schlüssels in Konto 2 (444455556666) auszuführen. Das `Action`-Element delegiert dem Prinzipal die gleichen Berechtigungen, die Konto 1 durch die Schlüsselrichtlinie in Konto 2 erhalten hat. Um diese Berechtigung für die `Engineering`-Rolle in Konto 1, ist diese Inline-Richtlinie eingebettet in der `Engineering`-Rolle.

  Kontoübergreifende IAM-Richtlinien wie diese werden erst dann wirksam, wenn die Schlüsselrichtlinie für den KMS-Schlüssel in Konto 2 dem Konto 1 die Berechtigung zur Verwendung des KMS-Schlüssels erteilt. Konto 1 kann seinen Prinzipalen außerdem nur Berechtigungen zum Ausführen von Aktionen erteilen, die dem Konto über die Schlüsselrichtlinie gewährt wurden.

  Im Flussdiagramm erfüllt dies die Bedingung Does an IAM policy allow the caller to perform this action? (Erlaubt eine IAM-Richtlinie dem Aufrufer die Ausführung dieser Aktion?).

```
{
```

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

- Das letzte erforderliche Element ist die Definition der `Engineering`-Rolle in Konto 1. Das `AssumeRolePolicyDocument` in der Rolle erlaubt Bob, die `Engineering`-Rolle zu übernehmen.

```
{
    "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 Berechtigungen

Diese Tabelle soll Ihnen helfen, die AWS KMS Berechtigungen zu verstehen, sodass Sie den Zugriff auf Ihre AWS KMS Ressourcen kontrollieren können. Definitionen der Spaltenüberschriften werden unter der Tabelle angezeigt.

Informationen zu AWS KMS Berechtigungen finden Sie auch im Abschnitt Aktionen, Ressourcen und Bedingungsschlüssel zum AWS Key Management Service Thema Service Authorization Reference. In diesem Thema werden jedoch nicht alle Bedingungsschlüssel aufgeführt, die Sie zum Verfeinern jeder Berechtigung verwenden können.

> **Note**
>
> Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in der Tabelle anzuzeigen.

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
CancelKeyDeletion `kms:CancelKeyDeletion`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
ConnectCustomKeyStore `kms:ConnectCustomKeyStore`	IAM-Richtlinie	Nein	*	km: CallerAccount

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
CreateAlias `kms:CreateAlias` Um diese Produktion verwenden zu können, benötigt der Aufrufer die Berechtigung `kms:CreateAlias` für zwei Ressourcen: • Alias (in einer IAM-Richtlinie) • Der KMS-Schlüssel (in einer Schlüsselrichtlinie) Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.	IAM-Richtlinie (für den Alias)	Nein	Alias	Keine (bei der Steuerung des Zugriffs auf den Alias)
	Schlüsselrichtlinie (für den KMS-Schlüssel)	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
CreateCustomKeyStore `kms:CreateCustomKeyStore`	IAM-Richtlinie	Nein	*	km: CallerAccount

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
CreateGrant `kms:CreateGrant`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für Erteilungen: km: GrantConstraintType km: GranteePrincipal km: GrantIsForAWSResource km: GrantOperations km: RetiringPrincipal Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
CreateKey `kms:CreateKey`	IAM-Richtlinie	Nein	*	km: BypassPolicyLockoutSafetyCheck km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ViaService aws:RequestTag/tag-key (AWS globaler Bedingungsschlüssel) aws:ResourceTag/tag-key (globaler Bedingungsschlüssel)AWS aws: TagKeys (AWS globaler Bedingungsschlüssel)

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
Decrypt `kms:Decrypt`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
DeleteAlias `kms:DeleteAlias`	IAM-Richtlinie (für den Alias)	Nein	Alias	Keine (bei der Steuerung des Zugriffs auf den Alias)
Um diese Produktion verwenden zu können, benötigt der Aufrufer die Berechtigung `kms:DeleteAlias` für zwei Ressourcen: • Alias (in einer IAM-Richtlinie) • Der KMS-Schlüssel (in einer Schlüsselrichtlinie) Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.	Schlüsselrichtlinie (für den KMS-Schlüssel)	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
DeleteCustomKeyStore `kms:DeleteCustomKeyStore`	IAM-Richtlinie	Nein	*	km: CallerAccount
DeleteImportedKeyMaterial `kms:DeleteImportedKeyMaterial`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
DescribeCustomKeyStores `kms:DescribeCustomKeyStores`	IAM-Richtlinie	Nein	*	km: CallerAccount

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
DescribeKey `kms:DescribeKey`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: RequestAlias

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
DisableKey `kms:DisableKey`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
DisableKeyRotation `kms:DisableKeyRotation`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
DisconnectCustomKeyStore `kms:DisconnectCustomKeyStore`	IAM-Richtlinie	Nein	*	km: CallerAccount

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
EnableKey `kms:EnableKey`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
EnableKeyRotation `kms:EnableKeyRotation`	Schlüsselrichtlinie	Nein	KMS-Schlüssel (nur symmetrisch)	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Bedingungen für die automatische Schlüsselrotation: km: RotationPeriodInDays

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
Encrypt `kms:Encrypt`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GenerateDataKey `kms:GenerateDataKey`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur symmetrisch)	Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GenerateDataKeyPair `kms:GenerateDataKeyPair`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur symmetrisch) Generierung eines asymmetrischen Datenschlüsselpaars, das durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung geschützt ist.	Bedingungen für Datenschlüsselpaare: km: DataKeyPairSpec Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GenerateDataKeyPairWithoutPlaintext `kms:GenerateDataKeyPairWithoutPlaintext`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur symmetrisch) Generierung eines asymmetrischen Datenschlüsselpaars, das durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung geschützt ist.	Bedingungen für Datenschlüsselpaare: km: DataKeyPairSpec Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GenerateDataKeyWithoutPlaintext `kms:GenerateDataKeyWithoutPlaintext`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur symmetrisch)	Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
GenerateMac `kms:GenerateMac`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Bedingungen für kryptografische Operationen km: MacAlgorithm km: RequestAlias

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GenerateRandom `kms:GenerateRandom`	IAM-Richtlinie	N/A	*	None
GetKeyPolicy `kms:GetKeyPolicy`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GetKeyRotationStatus `kms:GetKeyRotationStatus`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur symmetrisch)	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GetParametersForImport `kms:GetParametersForImport`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	km: WrappingAlgorithm km: WrappingKeySpec Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
GetPublicKey `kms:GetPublicKey`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur asymmetrisch)	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: RequestAlias

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ImportKeyMaterial `kms:ImportKeyMaterial`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: ExpirationModel km: ValidTo
ListAliases `kms:ListAliases`	IAM-Richtlinie	Nein	*	None

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ListGrants `kms:ListGrants`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: GrantIsForAWSResource

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ListKeyPolicies `kms:ListKeyPolicies`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ListKeyRotations `kms:ListKeyRotations`	Schlüsselrichtlinie	Nein	KMS-Schlüssel (nur symmetrisch)	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
ListKeys `kms:ListKeys`	IAM-Richtlinie	Nein	*	None

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ListResourceTags `kms:ListResourceTags`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ListRetirableGrants `kms:ListRetirableGrants`	IAM-Richtlinie	Der angegebene Prinzipal muss sich im lokalen Konto befinden, aber die Produktion gibt Erteilungen in allen Konten zurück.	*	None

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
PutKeyPolicy `kms:PutKeyPolicy`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: BypassPolicyLockoutSafetyCheck

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ReEncrypt kms:ReEncryptFrom kms:ReEncryptTo Um diese Produktion verwenden zu können, benötigt der Anrufer die Berechtigung für zwei KMS-Schlüssel: • kms:ReEncryptFrom auf dem KMS-Schlüssel zum Entschlüsseln • kms:ReEncryptTo auf dem KMS-Schlüssel zum Verschlüsseln	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für kryptografische Operationen km: EncryptionAlgorithm km: RequestAlias Bedingungen für den Verschlüsselungskontext: kms:EncryptionContext: Kontextschlüssel km: EncryptionContextKeys Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
				aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: ReEncryptOnSameKey

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ReplicateKey `kms:ReplicateKey` Um diese Produktion verwenden zu können, benötigt der Anrufer die folgenden Berechtigungen: • `kms:ReplicateKey` auf dem multiregionalen Primärschlüssel • `kms:CreateKey` in einer IAM-Richtlinie in der Replikatregion	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: ReplicaRegion

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
RetireGrant `kms:RetireGrant` Die Berechtigung zur Außerbetriebnahme einer Erteilung wird hauptsächlich durch die Erteilung bestimmt. Eine Richtlinie allein kann den Zugriff auf diese Produktion nicht erlauben. Weitere Informationen finden Sie unter Außerbetriebnahme und Widerruf von Erteilungen.	IAM-Richtlinie (Diese Berechtigung ist in einer Schlüsselrichtlinie nicht wirksam.)	Ja	KMS-Schlüssel	km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel)

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
RevokeGrant `kms:RevokeGrant`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen: km: GrantIsForAWSResource

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
RotateKeyOnDemand `kms:RotateKeyOnDemand`	Schlüsselrichtlinie	Nein	KMS-Schlüssel (nur symmetrisch)	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
ScheduleKeyDeletion `kms:ScheduleKeyDeletion`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
Sign `kms:Sign`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur asymmetrisch)	Bedingungen für Signatur und Verifizierung: km: MessageType km: RequestAlias km: SigningAlgorithm Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
TagResource `kms:TagResource`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Bedingungen für Markierung: aws:RequestTag/tag-key (AWS globaler Bedingungsschlüssel) aws: TagKeys (AWS globaler Bedingungsschlüssel)

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
UntagResource `kms:UntagResource`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Bedingungen für Markierung: aws:RequestTag/tag-key (AWS globaler Bedingungsschlüssel) aws: TagKeys (AWS globaler Bedingungsschlüssel)

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
UpdateAlias `kms:UpdateAlias` Um diese Produktion verwenden zu können, benötigt der Aufrufer die Berechtigung `kms:UpdateAlias` für drei Ressourcen: • Alias • Der aktuell zugeordnete KMS-Schlüssel • Der neu zugeordnete KMS-Schlüssel Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.	IAM-Richtlinie (für den Alias)	Nein	Alias	Keine (bei der Steuerung des Zugriffs auf den Alias)
	Schlüsselrichtlinie (für die KMS-Schlüssel)	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService
UpdateCustomKeyStore `kms:UpdateCustomKeyStore`	IAM-Richtlinie	Nein	*	km: CallerAccount

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
UpdateKeyDescription `kms:UpdateKeyDescription`	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
UpdatePrimaryRegion `kms:UpdatePrimaryRegion` Um diese Produktion verwenden zu können, benötigt der Aufrufer die `kms:UpdatePrimaryRegion` -Berechtigung sowohl für den multiregionalen Primärschlüssel, der zu einem Replikatschlüssel wird, und den multiregionalen Replikatschlüssel, der zum Primärschlüssel wird.	Schlüsselrichtlinie	Nein	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Andere Bedingungen km: PrimaryRegion

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
Verify `kms:Verify`	Schlüsselrichtlinie	Ja	KMS-Schlüssel (nur asymmetrisch)	Bedingungen für Signatur und Verifizierung: km: MessageType km: RequestAlias km: SigningAlgorithm Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService

Aktionen und Berechtigungen	Richtlinientyp	Kontoübergreifende Verwendung	Ressourcen (für IAM-Richtlinien)	AWS KMS Bedingungsschlüssel
VerifyMac `kms:VerifyMac`	Schlüsselrichtlinie	Ja	KMS-Schlüssel	Bedingungen für KMS-Schlüssel-Operationen: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (AWS globaler Bedingungsschlüssel) km: ViaService Bedingungen für kryptografische Operationen km: MacAlgorithm km: RequestAlias

Beschreibungen der Spalten

Die Spalten in dieser Tabelle enthalten folgende Informationen:

- Unter Aktionen und Berechtigungen werden alle AWS KMS API-Operationen und die Berechtigungen aufgeführt, die den Vorgang zulassen. Sie geben die Produktion im `Action`-Element einer Richtlinienanweisung an.
- Policy type (Richtlinientyp) gibt an, ob die Berechtigung in einer Schlüsselrichtlinie oder einer IAM-Richtlinie verwendet werden kann.

  Key policy (Schlüsselrichtlinie) bedeutet, dass Sie die Berechtigung in der Schlüsselrichtlinie angeben können. Wenn die Schlüsselrichtlinie die Richtlinienanweisung zur Aktivierung von IAM-Richtlinien enthält, können Sie die Berechtigung in einer IAM-Richtlinie angeben.

  IAM policy (IAM-Richtlinie) bedeutet, dass Sie die Berechtigung nur in einer IAM-Richtlinie angeben können.
- Kontenübergreifende Verwendung zeigt die Operationen an, die autorisierte Benutzer für Ressourcen in einem anderen AWS-Konto ausführen können.

  Ein Wert von Ja bedeutet, dass Prinzipale die Produktion für Ressourcen in einem anderen AWS-Konto ausführen können.

  Ein Wert von No (Nein) bedeutet, dass Prinzipale die Produktion nur auf Ressourcen in ihrem eigenen AWS-Konto ausführen können.

  Wenn Sie einem Prinzipal in einem anderen Konto eine Berechtigung erteilen, die nicht für eine kontoübergreifende Ressource verwendet werden kann, ist die Berechtigung nicht wirksam. Wenn Sie beispielsweise einem Prinzipal in einem anderen Konto die TagResource Berechtigung kms: für einen KMS-Schlüssel in Ihrem Konto erteilen, schlagen seine Versuche fehl, den KMS-Schlüssel in Ihrem Konto zu kennzeichnen.
- Resources listet die AWS KMS Ressourcen auf, für die die Berechtigungen gelten. AWS KMS unterstützt zwei Ressourcentypen: einen KMS-Schlüssel und einen Alias. In einer Schlüsselrichtlinie ist der Wert des `Resource`-Elements stets `*`. Dies gibt den KMS-Schlüssel an, dem die Schlüsselrichtlinie angefügt ist.

  Verwenden Sie die folgenden Werte, um eine AWS KMS Ressource in einer IAM-Richtlinie darzustellen.

  KMS-Schlüssel

  Wenn es sich bei der Ressource um einen KMS-Schlüssel handelt, verwenden Sie dessen Schlüssel-ARN. Weitere Informationen dazu finden Sie unter the section called “Finden der Schlüssel-ID und des Schlüssel-ARN”.

`arn:AWS_partition_name:kms:AWS_Region:AWS_account_ID:key/key_ID`

Beispielsweise:

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

Alias

Wenn es sich bei der Ressource um einen Alias handelt, verwenden Sie den Alias-ARN. Weitere Informationen dazu finden Sie unter the section called “Suchen des Aliasnamens und des Alias-ARN”.

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Beispielsweise:

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

***** (Sternchen)

Wenn die Berechtigung nicht für eine bestimmte Ressource (KMS-Schlüssel oder Alias) gilt, verwenden Sie ein Sternchen (`*`).

In einer IAM-Richtlinie für eine Berechtigung steht ein Sternchen im Element für alle Ressourcen (KMS-Schlüssel und Aliase). AWS KMS `Resource` AWS KMS Sie können in dem `Resource` Element auch ein Sternchen verwenden, wenn die AWS KMS Berechtigung nicht für bestimmte KMS-Schlüssel oder -Aliase gilt. Wenn Sie beispielsweise die Berechtigungen `kms:CreateKey` oder `kms:ListKeys` erlauben oder ablehnen, können Sie das Element `Resource` auf `*` oder auf eine kontospezifische Variante festlegen, z. B. `arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:*`.

- AWS KMS Bedingungsschlüssel listet die AWS KMS Bedingungsschlüssel auf, mit denen Sie den Zugriff auf den Vorgang steuern können. Sie geben Bedingungen im `Condition`-Element einer Richtlinie an. Weitere Informationen finden Sie unter AWS KMS Bedingungsschlüssel. Diese Spalte enthält auch AWS globale Bedingungsschlüssel, die von AWS KMS, aber nicht von allen AWS Diensten unterstützt werden.

Testen der Berechtigungen

Um AWS KMS zu verwenden, müssen Sie über Anmeldeinformationen verfügen, die AWS zur Authentifizierung Ihrer API-Anforderungen verwenden kann. Die Anmeldedaten müssen die Berechtigung zum Zugriff auf KMS-Schlüssel und Aliase enthalten. Die Berechtigungen werden durch Schlüsselrichtlinien, IAM-Richtlinien, Zuschüsse und kontoübergreifende Zugriffskontrollen bestimmt. Sie können nicht nur den Zugriff auf KMS-Schlüssel steuern, sondern auch den Zugriff auf Ihr CloudHSM und auf Ihre benutzerdefinierten Schlüsselspeicher.

Sie können den `DryRun`-API-Parameter angeben, um zu überprüfen, ob Sie über die erforderlichen Berechtigungen verfügen, um AWS KMS-Schlüssel zu verwenden. Sie können `DryRun` auch verwenden, um zu überprüfen, ob die Anforderungsparameter in einem AWS KMS-API-Aufruf korrekt angegeben sind.

Themen

Was ist der - DryRun Parameter?

`DryRun` ist ein optionaler API-Parameter, den Sie angeben, um zu überprüfen, ob AWS KMS-API-Aufrufe erfolgreich sind. Verwenden Sie `DryRun`, um Ihren API-Aufruf zu testen, bevor Sie AWS KMS tatsächlich aufrufen. Sie können die folgenden Punkte überprüfen.

- Dass Sie über die erforderlichen Berechtigungen verfügen, um AWS KMS-Schlüssel zu verwenden.
- Dass Sie die Parameter im Aufruf korrekt angegeben haben.

AWS KMS unterstützt die Verwendung des `DryRun`-Parameters in bestimmten API-Aktionen:

- CreateGrant
- Decrypt
- Encrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext

- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign
- Verify
- VerifyMac

Die Verwendung des `DryRun`-Parameters ist kostenpflichtig und wird als Standard-API-Anfrage in Rechnung gestellt. Weitere Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service – Preise.

Alle API-Anfragen, die den `DryRun`-Parameter verwenden, beziehen sich auf das Anforderungskontingent der API und können zu einer Drosselungsausnahme führen, wenn Sie ein API-Anforderungskontingent überschreiten. Beispielsweise wird der Aufruf von Decrypt mit `DryRun` oder ohne `DryRun` demselben Kontingent für kryptografische Operationen angerechnet. Weitere Informationen hierzu finden Sie unter Drosselung AWS KMS von Anfragen.

Jeder Aufruf an einen AWS KMS-API-Vorgang wird als Ereignis erfasst und in einem AWS CloudTrail-Protokoll aufgezeichnet. Die Ausgabe aller Operationen, die den `DryRun` Parameter angeben, wird in Ihrem CloudTrail Protokoll angezeigt. Weitere Informationen finden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

Angeben DryRun mit der API

Um `DryRun` zu verwenden, geben Sie den Parameter `–dry-run` in AWS CLI-Befehlen und AWS KMS-API-Aufrufen an, die den Parameter unterstützen. Wenn Sie dies tun, prüft AWS KMS, ob Ihr Aufruf erfolgreich sein wird. AWS KMS-Aufrufe, die `DryRun` verwenden, schlagen immer fehl und geben eine Meldung mit Informationen über den Grund für das Scheitern des Aufrufs zurück. Die Nachricht kann die folgenden Ausnahmen enthalten:

- `DryRunOperationException` – Die Anfrage wäre erfolgreich, wenn `DryRun` nicht angegeben wäre.
- `ValidationException` – Die Anfrage schlug fehl, weil ein falscher API-Parameter angegeben wurde.

- `AccessDeniedException` – Sie sind nicht berechtigt, die angegebene API-Aktion auf der KMS-Ressource auszuführen.

Der folgende Befehl verwendet beispielsweise die -CreateGrantOperation und erstellt eine Erteilung, die es Benutzern, die die `keyUserRole` Rolle annehmen dürfen, ermöglicht, die Decrypt-Operation für einen angegebenen symmetrischen KMS-Schlüssel aufzurufen. Der `DryRun`-Parameter ist angegeben.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

Schlüssel für spezielle Zwecke

AWS Key Management Service(AWS KMS) unterstützt verschiedene Arten von Schlüsseln für verschiedene Verwendungszwecke.

Wenn Sie einen AWS KMS key erstellen, erhalten Sie standardmäßig einen KMS-Schlüssel mit symmetrischer Verschlüsselung. In AWS KMS stellt ein KMS-Schlüssel mit symmetrischer Verschlüsselung einen 256-Bit-AES-GCM-Verschlüsselungsschlüssel dar, der für Verschlüsselung und Entschlüsselung verwendet wird, außer in den China-Regionen, in denen er einen symmetrischen 128-Bit-Schlüssel darstellt, der SM4-Verschlüsselung verwendet. Symmetrisches Schlüsselmaterial lässt zu keiner Zeit AWS KMS unverschlüsselt. Sofern Ihre Aufgabe nicht explizit asymmetrische Verschlüsselung oder HMAC-Schlüssel erfordert, sind KMS-Schlüssel mit symmetrischer Verschlüsselung, die AWS KMS niemals unverschlüsselt lassen, eine gute Wahl. Auch AWS-Services, die mit AWS KMS integriert sind, verwenden zum Verschlüsseln Ihrer Daten nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Diese Services unterstützen keine Verschlüsselung mit asymmetrischen KMS-Schlüsseln.

Sie können einen KMS-Schlüssel mit symmetrischer Verschlüsselung in AWS KMS verwenden, um Daten zu verschlüsseln, zu entschlüsseln und neu zu verschlüsseln, Datenschlüssel und Datenschlüsselpaare zu generieren und zufällige Byte-Zeichenfolgen zu generieren. Sie können in einen KMS-Schlüssel mit symmetrischer Verschlüsselung Ihr eigenes Schlüsselmaterial importieren und KMS-Schlüssel mit symmetrischer Verschlüsselung in benutzerdefinierten Schlüsselspeichern erstellen. Eine Tabelle mit den Operationen, die Sie für symmetrische und asymmetrische KMS-Schlüssel ausführen können, finden Sie unter Schlüsseltypreferenz.

AWS KMS unterstützt auch die folgenden speziellen KMS-Schlüsseltypen:

- Asymmetrische RSA-Schlüssel zur Kryptografie mit öffentlichen Schlüsseln
- Asymmetrische RSA- und ECC-Schlüssel zum Signieren und zur Verifizierung
- Asymmetrische SM2-Schlüssel (nur in den China-Regionen) für Kryptografie mit öffentlichen Schlüsseln oder Signieren und Verifizieren
- HMAC-Schlüssel, um Hash-basierte Nachrichtenauthentifizierungscodes zu generieren und zu überprüfen
- Multiregionale Schlüssel(symmetrisch und asymmetrisch), die wie Kopien desselben Schlüssels in verschiedenen AWS-Regionen funktionieren
- Schlüssel mit importiertem Schlüsselmaterial, die Sie bereitstellen

- Schlüssel in einem benutzerdefinierten Schlüsselspeicher, der von einem AWS CloudHSM-Cluster oder einem externen Schlüsselmanager außerhalb von AWS gestützt wird.

Auswahl eines KMS-Schlüsseltyps

AWS KMS unterstützt verschiedene Arten von KMS-Schlüsseln: symmetrische Verschlüsselungsschlüssel, symmetrische HMAC-Schlüssel, asymmetrische Verschlüsselungsschlüssel und asymmetrische Signaturschlüssel.

KMS-Schlüssel unterscheiden sich, da sie unterschiedliches kryptografisches Schlüsselmaterial enthalten.

- KMS-Schlüssel mit symmetrischer Verschlüsselung: Stellt einen 256-Bit-AES-GCM-Verschlüsselungsschlüssel dar, außer in den China-Regionen, in denen er einen 128-Bit-SM4-Verschlüsselungsschlüssel darstellt. Symmetrisches Schlüsselmaterial lässt zu keiner Zeit AWS KMS unverschlüsselt. Um Ihren KMS-Schlüssel mit symmetrischer Verschlüsselung zu verwenden, müssen Sie AWS KMS aufrufen.

  Symmetrische Verschlüsselungsschlüssel, bei denen es sich um die Standard-KMS-Schlüssel handelt, sind ideal für die meisten Anwendungen. Wenn Sie einen KMS-Schlüssel zum Schutz Ihrer Daten in einem AWS-Service benötigen, verwenden Sie einen symmetrischen Verschlüsselungsschlüssel, es sei denn, Sie werden angewiesen, einen anderen Schlüsseltyp zu verwenden.
- Asymmetrischer KMS-Schlüssel: repräsentiert ein Paar mathematisch verwandter Schlüssel, bestehend aus einem öffentlichen Schlüssel und einem privaten Schlüssel, das Sie für Verschlüsselung und Entschlüsselung oder Signierung und Verifizierung verwenden können, aber nicht für beides. Der private Schlüssel verlässt AWS KMS niemals unverschlüsselt. Sie können den öffentlichen Schlüssel innerhalb von AWS KMS verwenden, indem Sie die AWS KMS-API-Operationen aufrufen oder den öffentlichen Schlüssel herunterladen und ihn außerhalb von AWS KMS verwenden.
- HMAC-KMS-Schlüssel (symmetrisch): repräsentiert einen symmetrischen Schlüssel unterschiedlicher Länge, der zum Generieren und Überprüfen von Hash-basierten Nachrichtenauthentifizierungscodes verwendet wird. Das Schlüsselmaterial in einem HMAC-KMS-Schlüssel lässt AWS KMS zu keiner Zeit unverschlüsselt. Um Ihren HMAC-KMS-Schlüssel zu verwenden, müssen Sie AWS KMS aufrufen.

Welchen Typ von KMS-Schlüssel Sie erstellen, hängt in erster Linie von Ihrer beabsichtigten Verwendung des KMS-Schlüssels, Ihren Sicherheitsanforderungen und Ihren Autorisierungsanforderungen ab. Denken Sie beim Erstellen des KMS-Schlüssels daran, dass die kryptografische Konfiguration des KMS-Schlüssels, einschließlich der Schlüsselspezifikation und der Schlüsselnutzung, beim Erstellen des KMS-Schlüssels eingerichtet wird und nicht geändert werden kann.

Verwenden Sie die folgende Anleitung, um zu bestimmen, welchen Typ von KMS-Schlüssel Sie für Ihren Anwendungsfall benötigen.

Verschlüsseln und Entschlüsseln von Daten

Verwenden Sie einen symmetrischen KMS-Schlüssel für die meisten Anwendungsfälle, bei denen Daten verschlüsselt und entschlüsselt werden müssen. Der symmetrische Verschlüsselungsalgorithmus, den AWS KMS verwendet, ist schnell, effizient und gewährleistet die Vertraulichkeit und Authentizität von Daten. Er unterstützt authentifizierte Verschlüsselung mit zusätzlichen authentifizierten Daten (AAD), die als Verschlüsselungskontext definiert sind. Für diesen KMS-Schlüsseltyp müssen sowohl Absender als auch Empfänger verschlüsselter Daten über gültige AWS-Anmeldeinformationen verfügen, um AWS KMS aufrufen zu können.

Wenn Ihr Anwendungsfall eine Verschlüsselung außerhalb von AWS durch Benutzer erfordert, die AWS KMS nicht aufrufen können, sind asymmetrische KMS-Schlüssel eine gute Wahl. Sie können den öffentlichen Schlüssel des asymmetrischen KMS-Schlüssels verteilen, damit diese Benutzer Daten verschlüsseln können. Ihre Anwendungen, die diese Daten entschlüsseln müssen, können den privaten Schlüssel des asymmetrischen KMS-Schlüssels innerhalb von AWS KMS verwenden.

Signieren von Nachrichten und Verifizieren von Signaturen

Um Nachrichten zu signieren und Signaturen zu überprüfen, müssen Sie einen asymmetrischen KMS-Schlüssel verwenden. Sie können einen KMS-Schlüssel mit einer Schlüsselspezifikation verwenden, die ein RSA-Schlüsselpaar, ein ECC (Elliptic Curve)-Schlüsselpaar oder ein SM2-Schlüsselpaar (nur in den China-Regionen) darstellt. Die ausgewählte Schlüsselspezifikation wird durch den Signaturalgorithmus bestimmt, den Sie verwenden möchten. Die ECDSA-Signaturalgorithmen, die von ECC-Schlüsselpaaren unterstützt werden, sind den RSA-Signaturalgorithmen vorzuziehen. Möglicherweise müssen Sie jedoch eine bestimmte Schlüsselspezifikation und einen bestimmten Signaturalgorithmus verwenden, um Benutzer zu unterstützen, die Signaturen außerhalb von AWS überprüfen.

Verschlüsselung öffentlicher Schlüssel

Um die Verschlüsselung öffentlicher Schlüssel durchzuführen, müssen Sie einen asymmetrischen KMS-Schlüssel mit einer RSA-Schlüsselspezifikation oder einer SM2-Schlüsselspezifikation (nur in den China-Regionen) verwenden. Um Daten in AWS KMS mit dem öffentlichen Schlüssel eines KMS-Schlüsselpaars zu verschlüsseln, verwenden Sie die Encrypt-Operation. Sie können den öffentlichen Schlüssel auch herunterladen und für die Parteien freigeben, die Daten außerhalb von AWS KMS verschlüsseln müssen.

Wenn Sie den öffentlichen Schlüssel eines asymmetrischen KMS-Schlüssels herunterladen, können Sie ihn außerhalb von AWS KMS verwenden. Er unterliegt dann jedoch nicht mehr den Sicherheitskontrollen, die den KMS-Schlüssel in AWS KMS schützen. Sie können z. B. keine AWS KMS-Schlüsselrichtlinien oder Erteilungen verwenden, um die Verwendung des öffentlichen Schlüssels zu steuern. Sie können auch nicht steuern, ob der Schlüssel nur für die Verschlüsselung und Entschlüsselung mit den von AWS KMS unterstützten Verschlüsselungsalgorithmen verwendet wird. Weitere Informationen finden Sie unter Besondere Überlegungen zum Herunterladen öffentlicher Schlüssel.

Um Daten zu entschlüsseln, die mit dem öffentlichen Schlüssel außerhalb von AWS KMS verschlüsselt wurden, rufen Sie die Produktion Decrypt (Entschlüsseln) auf. Die `Decrypt`-Produktion schlägt fehl, wenn die Daten mit einem öffentlichen Schlüssel aus einem KMS-Schlüssel mit der `SIGN_VERIFY`-Schlüsselnutzung verschlüsselt wurden. Sie schlägt auch fehl, wenn sie mithilfe eines Algorithmus verschlüsselt wurde, den AWS KMS für die von Ihnen ausgewählte Schlüsselspezifikation nicht unterstützt. Weitere Informationen über wichtige Spezifikationen und unterstützte Algorithmen finden Sie unter Asymmetrische Schlüsselspezifikationen.

Um solche Fehler zu vermeiden, muss jeder, der einen öffentlichen Schlüssel außerhalb von AWS KMS verwendet, die Schlüsselkonfiguration speichern. Die AWS KMS Konsole und die GetPublicKey Antwort enthalten die Informationen, die Sie angeben müssen, wenn Sie den öffentlichen Schlüssel freigeben.

Generieren und überprüfen von HMAC-Codes.

Verwenden Sie einen HMAC-KMS-Schlüssel, um Hash-basierte Nachrichtenauthentifizierungscodes zu generieren und zu überprüfen. Wenn Sie einen HMAC-Schlüssel in AWS KMS erstellen, erstellt und schützt AWS KMS Ihr Schlüsselmaterial und stellt sicher, dass Sie die richtigen MAC-Algorithmen für Ihren Schlüssel verwenden. HMAC-Codes können auch als Pseudozufallszahlen und in bestimmten Szenarien zum symmetrischen Signieren und Tokenisieren verwendet werden.

HMAC-KMS-Schlüssel sind symmetrische Schlüssel. Beim Erstellen eines HMAC-KMS-Schlüssels in der AWS KMS-Konsole wählen Sie den Schlüsseltyp `Symmetric`.

Verwendung mit AWS-Services

Weitere Informationen zum Erstellen eines KMS-Schlüssel für die Verwendung mit einem AWS-Service, der in AWS KMS integriert ist, finden Sie in der Dokumentation für den Service. AWS-Services, die Ihre Daten verschlüsseln, erfordern einen KMS-Schlüssel mit symmetrischer Verschlüsselung.

Zusätzlich zu diesen Überlegungen haben kryptografische Operationen zu KMS-Schlüsseln mit unterschiedlichen Schlüsselspezifikationen unterschiedliche Preise und unterschiedliche Anforderungskontingente. Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service Pricing (Preise für WAF). Weitere Informationen zu Anforderungskontingenten finden Sie unter Anforderungskontingente.

Auswählen der Schlüsselnutzung

Die Schlüsselnutzung eines KMS-Schlüssels bestimmt, ob der KMS-Schlüssel für die Verschlüsselung und Entschlüsselung, Signierung und Verifizierung von Signaturen oder Generierung und Verifizierung von HMAC-Tags verwendet wird. Jeder KMS-Schlüssel kann nur eine Schlüsselverwendung haben. Die Verwendung eines KMS-Schlüssels für mehr als eine Art von Operation macht das Produkt aller Operationen anfälliger gegenüber Angriffen.

Wie in der folgenden Tabelle gezeigt, können KMS-Schlüssel mit symmetrischer Verschlüsselung nur für Verschlüsselung und Entschlüsselung verwendet werden. HMAC-KMS-Schlüssel können nur zum Generieren und Verifizieren von HMAC-Codes verwendet werden. Elliptic-Curve-KMS-Schlüssel (ECC) können nur für Signatur und Verifizierung verwendet werden. Sie müssen eine Entscheidung zur Schlüsselverwendung für RSA-KMS-Schlüssel treffen.

Gültige Schlüsselnutzung für KMS-Schlüsseltypen

KMS-Schlüsseltyp	Verschlüsseln und Entschlüsseln ENCRYPT_DECRYPT	Signieren und überprüfen SIGN_VERIFY	MAC generieren und verifizieren GENERATE_VERIFY_MAC
KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	⊗	⊗
HMAC-KMS-Schlüssel (symmetrisch)	⊗	⊗	✓
Asymmetrische KMS-Schlüssel mit RSA-Schlüsselpaaren	✓	✓	⊗
Asymmetrische KMS-Schlüssel mit ECC-Schlüsselpaaren	⊗	✓	⊗
Asymmetrische KMS-Schlüssel mit SM2-Schlüsselpaaren (nur China-Regionen)	✓	✓	⊗

In der AWS KMS-Konsole wählen Sie zuerst den Schlüsseltyp (symmetrisch oder asymmetrisch) und dann die Schlüsselverwendung aus. Der ausgewählte Schlüsseltyp bestimmt, welche Optionen der Schlüsselverwendung angezeigt werden. Die ausgewählte Schlüsselverwendung bestimmt, welche Schlüsselspezifikationen, falls zutreffend, angezeigt werden.

So wählen Sie eine Schlüsselnutzung in der AWS KMS-Konsole aus:

- Für KMS-Schlüssel mit symmetrischer Verschlüsselung (Standard) wählen Sie Verschlüsselung und Entschlüsselung.

- Wählen Sie für HMAC-KMS-Schlüssel Generate and verify MAC (MAC generieren und überprüfen) aus.
- Wählen Sie für asymmetrische KMS-Schlüssel mit Elliptic-Curve-Schlüsselmaterial (ECC) die Option Sign and verify (Signieren und Überprüfen).
- Wählen Sie für asymmetrische KMS-Schlüssel mit RSA-Schlüsselmaterial Encrypt and decrypt (Verschlüsseln und Entschlüsseln)oder Sign and verify (Signieren und Überprüfen).
- Wählen Sie für asymmetrische KMS-Schlüssel mit SM2-Schlüsselmaterial Encrypt and decrypt (Verschlüsseln und Entschlüsseln) oder Sign and verify (Signieren und Überprüfen) aus. Die SM2-Schlüsselspezifikation ist nur in den China-Regionen verfügbar.

Um es Prinzipalen zu erlauben, KMS-Schlüssel nur für eine bestimmte Schlüsselnutzung zu erstellen, verwenden Sie den kms:KeyUsage-Bedingungsschlüssel. Sie können mit dem Bedingungsschlüssel `kms:KeyUsage` Prinzipalen auch, basierend auf seiner Schlüsselnutzung, den Aufruf von API-Operationen für einen KMS-Schlüssel gewähren. Beispielsweise können Sie die Berechtigung zum Deaktivieren eines KMS-Schlüssels nur dann zulassen, wenn die Schlüsselnutzung SIGN_VERIFY lautet.

Auswählen der Schlüsselspezifikation

Wenn Sie einen asymmetrischen KMS-Schlüssel oder HMAC-KMS-Schlüssel erstellen, wählen Sie dessen Schlüsselspezifikation aus. Die Schlüsselspezifikation, die eine Eigenschaft jedes AWS KMS key ist, repräsentiert die kryptografische Konfiguration Ihres KMS-Schlüssels. Sie wählen die Schlüsselspezifikation, wenn Sie den KMS-Schlüssel erstellen. Sie kann danach nicht mehr geändert werden. Wenn Sie die falsche Schlüsselspezifikation ausgewählt haben, löschen Sie den KMS-Schlüssel und erstellen Sie einen neuen.

> **Note**
>
> Die Schlüsselspezifikation für einen KMS-Schlüssel wurde als „Kunden-Hauptschlüssel-Spezifikation“ bezeichnet. Der -`CustomerMasterKeySpec`Parameter der -CreateKeyOperation ist veraltet. Verwenden Sie stattdessen den `KeySpec`-Parameter. Die Antwort der - `CreateKey` und -DescribeKeyOperationen umfasst ein - `KeySpec` und -`CustomerMasterKeySpec`Element mit demselben Wert.

Die Schlüsselspezifikation bestimmt, ob der KMS-Schlüssel symmetrisch oder asymmetrisch ist, den Typ des Schlüsselmaterials im KMS-Schlüssel und die Verschlüsselungsalgorithmen, Signaturalgorithmen oder Nachrichten-Authentifizierungscode-Algorithmen (MAC) die AWS KMS für den KMS-Schlüssel unterstützt. Die von Ihnen gewählte Schlüsselspezifikation wird in der Regel durch Ihren Anwendungsfall und gesetzliche Anforderungen bestimmt. Allerdings haben kryptografische Operationen zu KMS-Schlüsseln mit unterschiedlichen Schlüsselspezifikationen unterschiedliche Preise und unterliegen unterschiedlichen Kontingenten. Details zu den Preisen finden Sie unter AWS Key Management Service-Preise. Weitere Informationen zu Anforderungskontingenten finden Sie unter Anforderungskontingente.

Um die Schlüsselspezifikationen zu ermitteln, die Prinzipale in Ihrem Konto für KMS-Schlüssel verwenden dürfen, verwenden Sie den kms:KeySpec-Bedingungsschlüssel.

AWS KMS unterstützt die folgenden Schlüsselspezifikationen für KMS-Schlüssel:

Symmetrische Verschlüsselungsschlüsselspezifikation(Standard)

- SYMMETRIC_DEFAULT

HMAC-Schlüsselspezifikationen

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

RSA-Schlüsselspezifikationen (Verschlüsselung und Entschlüsselung -oder- Signatur und Verifizierung)

- RSA_2048
- RSA_3072
- RSA_4096

Elliptic Curve(EC)-Schlüsselspezifikationen

- Asymmetrische NIST-empfohlene Elliptic Curve-Schlüsselpaare (Signatur und Verifizierung)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Andere asymmetrische Elliptic Curve-Schlüsselpaare (Signatur und Verifizierung)
  - ECC_SECG_P256K1 (secp256k1), häufig für Kryptowährung verwendet.

SM2-Schlüsselspezifikationen (Verschlüsselung und Entschlüsselung -oder- Signieren und Überprüfen)

- SM2 (nur China-Regionen)

Asymmetrische Schlüssel in AWS KMS

AWS KMS unterstützt asymmetrische KMS-Schlüssel, die ein mathematisch verwandtes öffentliches und privates RSA-, Elliptic Curve (ECC)- oder SM2 (nur Chine-Regionen)-Schlüsselpaar darstellen. Diese Schlüsselpaare werden in AWS KMS-Hardwaresicherheitsmodulen generiert, die unter dem FIPS 140-2 Cryptographic Module Validation Program zertifiziert sind, außer in den Regionen China (Peking) und China (Ningxia). Der private Schlüssel hinterlässt die AWS KMS HSMs niemals unverschlüsselt. Sie können den öffentlichen Schlüssel zur Verteilung und Verwendung außerhalb von AWS herunterladen. Sie können asymmetrische KMS-Schlüssel zur Verschlüsselung und Entschlüsselung oder zum Signieren und Verifizieren verwenden, jedoch nicht für beides.

Sie können die asymmetrischen KMS-Schlüssel in Ihrem AWS-Konto erstellen und verwalten und die Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen festlegen, die den Zugriff auf die steuern. Sie können die KMS-Schlüssel auch aktivieren und deaktivieren, Tags und Aliase erstellen sowie die KMS-Schlüssel löschen. Sie können alle Operationen, die Ihre asymmetrischen KMS-Schlüssel nutzen oder verwalten, in AWS in AWS CloudTrail-Protokollen überwachen.

AWS KMS bietet auch asymmetrische Datenschlüsselpaare, die zur clientseitigen Kryptografie außerhalb von AWS KMS entwickelt wurden. Der private Schlüssel in einem asymmetrischen Datenschlüsselpaar wird durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung in AWS KMS geschützt.

In diesem Thema wird erläutert, wie asymmetrische KMS-Schlüssel funktionieren, wie sie sich von anderen KMS-Schlüsseln unterscheiden und wie Sie entscheiden, welche Art von KMS-Schlüssel Sie zum Schutz Ihrer Daten benötigen. Außerdem wird erläutert, wie asymmetrische Datenschlüsselpaare funktionieren und wie sie außerhalb von AWS KMS verwendet werden.

Regionen

Asymmetrische KMS-Schlüssel und asymmetrische Datenschlüsselpaare werden in allen AWS-Regionen unterstützt, die AWS KMS unterstützt.

Weitere Informationen

- Informationen zum Erstellen asymmetrischer KMS-Schlüssel finden Sie unter Erstellen asymmetrischer KMS-Schlüssel. Informationen zum Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung finden Sie unter Erstellen von Schlüsseln.
- Informationen zum Erstellen multiregionaler asymmetrischer KMS-Schlüssel finden Sie unter Erstellen von multiregionalen Schlüsseln.
- Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.
- Eine Tabelle mit einem Vergleich der AWS KMS-API-Operationen, die auf die einzelnen KMS-Schlüsseltypen zutreffen, finden Sie unter the section called “Schlüsseltypreferenz”.
- Informationen zur Steuerung des Zugriffs auf die Schlüsselspezifikationen, Schlüsselnutzung, Verschlüsselungsalgorithmen und Signaturalgorithmen, die Prinzipale in Ihrem Konto für KMS-Schlüssel verwenden können, finden Sie unter the section called “AWS KMS Bedingungsschlüssel”.
- Weitere Informationen zu den Anforderungskontingenten für unterschiedliche KMS-Schlüsseltypen finden Sie unter the section called “Anforderungskontingente”.
- Informationen zum Signieren von Nachrichten und Überprüfen von Signaturen mit asymmetrischen KMS-Schlüsseln finden Sie unter Digitale Signierung mit dem neuen asymmetrischen Schlüssel-Feature von AWS KMS im AWS-Sicherheits-Blog.

Themen

Asymmetrische KMS-Schlüssel

Sie können einen asymmetrischen KMS-Schlüssel in AWS KMS erstellen. Ein asymmetrischer KMS-Schlüssel repräsentiert einen mathematisch verwandtes Schlüsselpaar mit einem öffentlichen und einem privaten Schlüssel. Sie können den öffentlichen Schlüssel jedem geben, auch wenn er nicht vertrauenswürdig sind, aber der private Schlüssel muss geheim gehalten werden.

In einem asymmetrischen KMS-Schlüssel wird der private Schlüssel in AWS KMS erstellt und verlässt AWS KMS niemals unverschlüsselt. Um den privaten Schlüssel zu verwenden, müssen Sie AWS KMS aufrufen. Sie können den öffentlichen Schlüssel innerhalb von AWS KMS verwenden, indem Sie die AWS KMS-API-Operationen aufrufen. Oder Sie können den öffentlichen Schlüssel herunterladen und ihn außerhalb von AWS KMS verwenden.

Wenn Ihr Anwendungsfall eine Verschlüsselung außerhalb von AWS durch Benutzer erfordert, die AWS KMS nicht aufrufen können, sind asymmetrische KMS-Schlüssel eine gute Wahl. Wenn Sie jedoch einen KMS-Schlüssel erstellen, um die Daten zu verschlüsseln, die Sie in einem AWS-Service speichern oder verwalten, verwenden Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung. AWS-Services, die in AWS KMS integriert sind, verwenden nur KMS-Schlüssel mit symmetrischer Verschlüsselung, um Ihre Daten zu verschlüsseln. Diese Services unterstützen keine Verschlüsselung mit asymmetrischen KMS-Schlüsseln.

AWS KMS unterstützt drei Arten von asymmetrischen KMS-Schlüsseln.

- RSA-KMS-Schlüssel: Ein KMS-Schlüssel mit einem RSA-Schlüsselpaar für Verschlüsselung und Entschlüsselung oder Signatur und Verifizierung (jedoch nicht für beides). AWS KMS unterstützt mehrere Schlüssellängen für unterschiedliche Sicherheitsanforderungen.
- Elliptc-Curve-KMS-Schlüssel (ECC): Ein KMS-Schlüssel mit einem Elliptic-Curve-Schlüsselpaar für Signatur und Verifizierung. AWS KMS unterstützt mehrere häufig verwendete Curves.
- SM2-KMS-Schlüssel (nur China-Regionen): Ein KMS-Schlüssel mit einem SM2-Schlüsselpaar für Verschlüsselung und Entschlüsselung oder Signatur und Überprüfung (aber nicht für beides).

Hilfe bei der Auswahl Ihrer asymmetrischen Schlüsselkonfiguration finden Sie unter Auswahl eines KMS-Schlüsseltyps. Technische Details zu den Verschlüsselungs- und Signaturalgorithmen, die AWS KMS für RSA-KMS-Schlüssel unterstützt, finden Sie unter RSA-Schlüsselspezifikationen. Technische Details zu den Signaturalgorithmen, die AWS KMS für ECC-KMS-Schlüssel unterstützt, finden Sie unter Elliptic-Curve-Schlüsselspezifikationen. Technische Details zu den Verschlüsselungs- und Signaturalgorithmen, die AWS KMS für SM2-KMS-Schlüssel (nur China-Regionen) unterstützt, finden Sie unter SM2-Schlüsselspezifikationen.

Eine Tabelle mit den Operationen, die Sie für symmetrische und asymmetrische KMS-Schlüssel ausführen können, finden Sie unter Vergleich symmetrischer und asymmetrischer KMS-Schlüssel. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Regionen

Asymmetrische KMS-Schlüssel und asymmetrische Datenschlüsselpaare werden in allen AWS-Regionen unterstützt, die AWS KMS unterstützt.

Erstellen asymmetrischer KMS-Schlüssel

Sie können asymmetrische KMS-Schlüssel in der -AWS KMSKonsole, mithilfe der CreateKey-API oder mithilfe einer AWS CloudFormation -Vorlage erstellen. Ein asymmetrischer KMS-Schlüssel repräsentiert ein Schlüsselpaar aus einem öffentlichen und einem privaten Schlüssel, das zur Verschlüsselung oder Signierung verwendet werden kann. Der private Schlüssel bleibt innerhalb AWS KMS. Informationen, um den öffentlichen Schlüssel zur Verwendung außerhalb von AWS KMS herunterzuladen, finden Sie unter Herunterladen öffentlicher Schlüssel.

Wenn Sie einen KMS-Schlüssel erstellen, um Daten zu verschlüsseln, die Sie in einem AWS-Service speichern oder verwalten, verwenden Sie einen KMS-Schlüssel zur symmetrischen Verschlüsselung. AWS-Services, die mit AWS KMS integriert werden, unterstützen keine asymmetrischen KMS-Schlüssel. Hilfe bei der Entscheidung, ob ein symmetrischer oder asymmetrischer KMS-Schlüssel erstellt werden soll, finden Sie unter Auswahl eines KMS-Schlüsseltyps.

Weitere Informationen über die Berechtigungen, die zum Erstellen von KMS-Schlüsseln erforderlich sind, finden Sie unter Berechtigungen zum Erstellen von KMS-Schlüsseln.

Themen

- Erstellen asymmetrischer KMS-Schlüssel (Konsole)
- Erstellen von symmetrischen KMS-Schlüsseln (AWS KMS-API)

Erstellen asymmetrischer KMS-Schlüssel (Konsole)

Sie können die AWS Management Console zum Erstellen von asymmetrischen AWS KMS keys (KMS-Schlüsseln) verwenden. Jeder asymmetrische KMS-Schlüssel repräsentiert ein Schlüsselpaar aus einem öffentlichen und einem privaten Schlüssel.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Um einen asymmetrischen KMS-Schlüssel zu erstellen, wählen Sie unter Key type (Schlüsseltyp) die Option Asymmetric (Asymmetrisch) aus.

   Hinweise zum Erstellen eines KMS-Schlüssels zur symmetrischen Verschlüsselung in der AWS KMS-Konsole finden Sie unter Erstellen von KMS-Schlüsseln mit symmetrischer Verschlüsselung (Konsole).
6. Um einen asymmetrischen KMS-Schlüssel für die Verschlüsselung öffentlicher Schlüssel zu erstellen, wählen Sie unter Key usage (Schlüsselnutzung) Encrypt and decrypt (Verschlüsseln und Entschlüsseln) aus. Um einen asymmetrischen KMS-Schlüssel zum Signieren von Nachrichten und zur Überprüfung von Signaturen zu erstellen, wählen Sie unter Key usage (Schlüsselnutzung) die Option Sign und verify (Signieren und Überprüfen) aus.

   Hilfe bei der Auswahl eines Schlüsselnutzungswerts finden Sie unter Auswählen der Schlüsselnutzung.
7. Wählen Sie eine Spezifikation (Key spec (Schlüsselspezifikation)) für Ihren asymmetrischen KMS-Schlüssel aus.

   Häufig wird die von Ihnen ausgewählte Schlüsselspezifikation durch gesetzliche, Sicherheits- oder geschäftliche Anforderungen bestimmt. Sie kann auch von der Größe der Nachrichten beeinflusst werden, die Sie verschlüsseln oder signieren müssen. Im Allgemeinen sind längere Verschlüsselungsschlüssel Brute-Force-Angriffen gegenüber weniger anfällig.

   Hilfe bei der Auswahl einer Schlüsselspezifikation finden Sie unter Auswählen der Schlüsselspezifikation.
8. Wählen Sie Next (Weiter).
9. Geben Sie einen Alias für den KMS-Schlüssel ein. Der Aliasname darf nicht mit **`aws/`** beginnen. Das Präfix **`aws/`** ist von Amazon Web Services reserviert und steht für Von AWS verwaltete Schlüssel in Ihrem Konto.

   Ein Alias ist ein Anzeigename, den Sie verwenden können, um den KMS-Schlüssel in der Konsole und in einigen AWS KMS-APIs zu identifizieren. Wir empfehlen, dass Sie einen Alias

wählen, der auf die Art von Daten, die Sie schützen möchten, oder die Anwendung, die Sie mit dem KMS-Schlüssel verwenden möchten, hindeutet.

Zum Erstellen eines KMS-Schlüssels in der Konsole benötigen Sie Aliase AWS Management Console. Sie können keinen Alias angeben, wenn Sie die -CreateKeyOperation verwenden, aber Sie können die Konsole oder die -CreateAliasOperation verwenden, um einen Alias für einen vorhandenen KMS-Schlüssel zu erstellen. Details hierzu finden Sie unter Verwenden von Aliassen.

10. (Optional) Geben Sie eine Beschreibung für den KMS-Schlüssel ein.

    Geben Sie eine Beschreibung ein, die die Art von Daten, die Sie schützen möchten, oder die Anwendung, die Sie mit dem KMS-Schlüssel verwenden möchten, erklärt.

    Sie können jetzt eine Beschreibung hinzufügen oder sie jederzeit aktualisieren, es sei denn, der Schlüsselstatus lautet `Pending Deletion` oder `Pending Replica Deletion`. Um die Beschreibung eines vorhandenen kundenverwalteten Schlüssels hinzuzufügen, zu ändern oder zu löschen, bearbeiten Sie die Beschreibung in der AWS Management Console oder verwenden Sie die -UpdateKeyDescriptionOperation.

11. (Optional) Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wählen Sie Add tag (Tag hinzufügen), wenn Sie mehr als ein Tag zum KMS-Schlüssel hinzufügen möchten.

    Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

12. Wählen Sie Weiter aus.

13. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel verwalten können.

> **Note**
>
> Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung zum Verwalten des KMS-Schlüssels erteilen. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre

Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

14. (Optional) Um zu verhindern, dass die ausgewählten IAM-Benutzer und -Rollen diesen KMS-Schlüssel löschen, deaktivieren Sie unten auf der Seite im Abschnitt Key deletion (Schlüssellöschung) das Kontrollkästchen Allow key administrators to delete this key (Administratoren erlauben, diesen Schlüssel zu löschen).
15. Wählen Sie Weiter aus.
16. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel für kryptographische Operationen verwenden können.

> **Note**
>
> Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung erteilen, den KMS-Schlüssel in kryptografischen Operationen zu verwenden. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

17. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other AWS-Konten (Andere Konten) unten auf der Seite die Option Add another AWS-Konto (Weiteres Konto hinzufügen) und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

> **Note**
>
> Um auch Prinzipalen aus den externen Konten Zugriff auf den KMS-Schlüssel zu erlauben, müssen die Administratoren der externen Konten IAM-Richtlinien erstellen, die diese Berechtigungen bereitstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

18. Wählen Sie Weiter.

19. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
20. Wählen Sie Finish (fertigstellen) aus, um den KMS-Schlüssel zu erstellen.

Erstellen von symmetrischen KMS-Schlüsseln (AWS KMS-API)

Sie können die -CreateKeyOperation verwenden, um einen asymmetrischen zu erstellenAWS KMS key. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Wenn Sie einen asymmetrischen KMS-Schlüssel erstellen, müssen Sie den `KeySpec`-Parameter angeben, der den Typ der von Ihnen erstellten Schlüssel bestimmt. Außerdem müssen Sie den `KeyUsage`-Wert ENCRYPT_DECRYPT oder SIGN_VERIFY angeben. Diese Eigenschaften können nicht geändert werden, nachdem der KMS-Schlüssel erstellt wurde.

Mit der -`CreateKey`Operation können Sie keinen Alias angeben, aber Sie können die -CreateAliasOperation verwenden, um einen Alias für Ihren neuen KMS-Schlüssel zu erstellen.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Im folgenden Beispiel wird die `CreateKey`-Produktion verwendet, um einen asymmetrischen KMS-Schlüssel von 4096-Bit-RSA-Schlüsseln für die Verschlüsselung öffentlicher Schlüssel zu erstellen.

```
$ 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
    }
}
```

Mit dem folgenden Beispielbefehl wird ein asymmetrischer KMS-Schlüssel erstellt, der ein Paar von ECDSA-Schlüssel repräsentiert, das für Signatur und Verifizierung verwendet wird. Sie können kein Elliptic Curve-Schlüsselpaar für die Verschlüsselung und Entschlüsselung erstellen.

```
$ 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"
    }
}
```

Herunterladen öffentlicher Schlüssel

Sie können den öffentlichen Schlüssel von einem asymmetrischen KMS-Schlüsselpaar mit der AWS Management Console- oder AWS KMS-API anzeigen, kopieren und herunterladen. Sie müssen über die `kms:GetPublicKey`-Berechtigung für den asymmetrischen KMS-Schlüssel verfügen.

Jedes asymmetrische KMS-Schlüsselpaar besteht aus einem privaten Schlüssel, der AWS KMS niemals unverschlüsselt verlässt, und einem öffentlichen Schlüssel, den Sie herunterladen und freigeben können.

Sie geben möglicherweise einen öffentlichen Schlüssel frei, um es anderen zu ermöglichen, Daten außerhalb von AWS KMS zu verschlüsseln, die Sie dann nur mit Ihrem privaten Schlüssel entschlüsseln können. Oder, um anderen zu erlauben, eine digitale Signatur, die Sie mit Ihrem privaten Schlüssel generiert haben, außerhalb von AWS KMS zu überprüfen.

Wenn Sie den öffentlichen Schlüssel in Ihrem asymmetrischen KMS-Schlüssel innerhalb von AWS KMS verwenden, profitieren Sie von der Authentifizierung, Autorisierung und Protokollierung, die Teil jeder AWS KMS-Produktion sind. Außerdem reduzieren Sie das Risiko, Daten zu verschlüsseln, die nicht entschlüsselt werden können. Diese Funktionen sind außerhalb von AWS KMS nicht wirksam. Details hierzu finden Sie unter Besondere Überlegungen zum Herunterladen öffentlicher Schlüssel.

> **Tip**
>
> Suchen Sie nach Datenschlüsseln oder SSH-Schlüsseln? In diesem Thema wird erläutert, wie Sie asymmetrische Schlüssel in AWS Key Management Service verwalten, bei denen der private Schlüssel nicht exportierbar ist. Informationen zu exportierbaren Datenschlüsselpaaren, bei denen der private Schlüssel durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung geschützt ist, finden Sie unter GenerateDataKeyPair. Hilfe beim Herunterladen des öffentlichen Schlüssels, der einer Amazon-EC2-Instance zugeordnet ist, finden Sie unter Abrufen des öffentlichen Schlüssels im Amazon-EC2-Benutzerhandbuch für Linux-Instances und Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Themen

Besondere Überlegungen zum Herunterladen öffentlicher Schlüssel

Um Ihre KMS-Schlüssel zu schützen, bietet AWS KMS Zugriffssteuerung, authentifizierte Verschlüsselung und detaillierte Protokolle jeder Produktion. AWS KMS erlaubt es Ihnen auch, die Verwendung von KMS-Schlüssel vorübergehend oder dauerhaft zu verhindern. Schließlich soll mit AWS KMS-Operationen das Risiko minimiert werden, dass Daten verschlüsselt werden, die nicht entschlüsselt werden können. Diese Funktionen sind nicht verfügbar, wenn Sie heruntergeladene öffentliche Schlüssel außerhalb von AWS KMS verwenden.

Autorisierung

Schlüsselrichtlinien und IAM-Richtlinien, die den Zugriff auf den KMS-Schlüssel in AWS KMS steuern, haben keine Auswirkungen auf Operationen, die außerhalb von AWS ausgeführt werden. Jeder Benutzer, der den öffentlichen Schlüssel abrufen kann, kann ihn außerhalb von AWS KMS verwenden, selbst wenn er nicht über die Berechtigung zum Verschlüsseln von Daten oder zur Überprüfung von Signaturen mit dem KMS-Schlüssel verfügt.

Nutzungsbeschränkungen für Schlüssel

Schlüssel-Nutzungsbeschränkungen sind außerhalb von AWS KMS nicht wirksam. Wenn Sie die Encrypt-Produktion mit einem KMS-Schlüssel aufrufen, dessen `KeyUsage`-Wert `SIGN_VERIFY` ist, schlägt die AWS KMS-Produktion fehl. Wenn Sie jedoch Daten außerhalb von AWS KMS mit einem öffentlichen Schlüssel von einem KMS-Schlüssel mit einem `KeyUsage`-Wert von `SIGN_VERIFY` verschlüsseln, können die Daten nicht entschlüsselt werden.

Algorithmusbeschränkungen

Beschränkungen für Verschlüsselungs- und Signaturalgorithmen, die von AWS KMS unterstützt werden, sind außerhalb von AWS KMS nicht wirksam. Wenn Sie Daten mit dem öffentlichen Schlüssel von einem KMS-Schlüssel außerhalb von AWS KMS verschlüsseln und einen Verschlüsselungsalgorithmus verwenden, den AWS KMS nicht unterstützt, können die Daten nicht entschlüsselt werden.

Deaktivieren und Löschen von KMS-Schlüsseln

Maßnahmen, die Sie ergreifen können, um die Verwendung des KMS-Schlüssel in einer kryptografischen Produktion innerhalb von AWS KMS zu verhindern, hindern niemanden daran, den öffentlichen Schlüssel außerhalb von AWS KMS zu verwenden. Beispielsweise hat das Deaktivieren eines KMS-Schlüssels, das Planen des Löschens eines KMS-Schlüssels, das Löschen eines KMS-Schlüssels oder das Löschen des Schlüsselmaterials aus einem KMS-Schlüssel keine Auswirkungen auf einen öffentlichen Schlüssel außerhalb von AWS KMS. Wenn

Sie einen asymmetrischen KMS-Schlüssel löschen oder dessen Schlüsselmaterial löschen oder verlieren, können Daten, die Sie mit einem öffentlichen Schlüssel außerhalb von AWS KMS verschlüsseln, nicht wiederhergestellt werden.

Protokollierung

AWS CloudTrail-Protokolle, die jede AWS KMS-Produktion, einschließlich Anforderung, Antwort, Datum, Uhrzeit und autorisierten Benutzer aufzeichnen, halten nicht die Verwendung des öffentlichen Schlüssels außerhalb von AWS KMS fest.

Offline-Überprüfung mit SM2-Schlüsselpaaren (nur China-Regionen)

Um eine Signatur außerhalb von AWS KMS mit einem öffentlichen SM2-Schlüssel zu überprüfen, müssen Sie die unterscheidende ID angeben. AWS KMS verwendet standardmäßig `1234567812345678` als die unterscheidende ID. Weitere Informationen finden Sie unter Offline-Überprüfung mit SM2-Schlüsselpaaren (nur China-Regionen).

Herunterladen eines öffentlichen Schlüssels (Konsole)

Sie können die AWS Management Console verwenden, um den öffentlichen Schlüssel von einem asymmetrischen KMS-Schlüssel in Ihrem AWS-Konto anzuzeigen, zu kopieren und herunterzuladen. Um den öffentlichen Schlüssel von einem asymmetrischen KMS-Schlüssel in einem anderen AWS-Konto herunterzuladen, verwenden Sie die AWS KMS API.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie den Alias oder die Schlüssel-ID eines asymmetrischen KMS-Schlüssels aus.
5. Wählen Sie die Registerkarte Cryptographic configuration (kryptografische Konfiguration) aus. Notieren Sie sich die Werte der Felder Key spec (Schlüsselspezifikation), Key usage (Schlüsselnutzung) und Encryption algorithms (Verschlüsselungsalgorithmen) oder Signing Algorithms (Signaturalgorithmen). Sie müssen diese Werte verwenden, um den öffentlichen Schlüssel außerhalb von AWS KMS zu verwenden. Stellen Sie sicher, dass Sie diese Informationen freigeben, wenn Sie den öffentlichen Schlüssel freigeben.
6. Wählen Sie die Registerkarte Public key (Öffentlicher Schlüssel).

7. Um den öffentlichen Schlüssel in die Zwischenablage zu kopieren, wählen Sie Copy (Kopieren). Um den öffentlichen Schlüssel in eine Datei herunterzuladen, wählen Sie Download (Herunterladen).

Herunterladen eines öffentlichen Schlüssels (AWS KMS-API)

Die GetPublicKey Operation gibt den öffentlichen Schlüssel in einem asymmetrischen KMS-Schlüssel zurück. Sie gibt auch wichtige Informationen zurück, die Sie benötigen, um den öffentlichen Schlüssel außerhalb von AWS KMS korrekt zu verwenden, einschließlich der Schlüsselnutzung und der Verschlüsselungsalgorithmen. Achten Sie darauf, diese Werte zu speichern und sie freizugeben, wenn Sie den öffentlichen Schlüssel freigeben.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Um einen KMS-Schlüssel anzugeben, verwenden Sie seine Schlüssel-ID, seinen Schlüssel-ARN, seinen Aliasnamen oder seinen Alias-ARN. Wenn Sie einen Aliasnamen verwenden, stellen Sie ihm alias/ voran. Um einen KMS-Schlüssel in einem anderen AWS-Konto anzugeben, müssen Sie seinen Schlüssel-ARN oder Alias-ARN verwenden.

Bevor Sie diesen Befehl ausführen, ersetzen Sie den Beispiel-Aliasnamen durch einen gültigen Bezeichner für den KMS-Schlüssel. Um diesen Befehl auszuführen, müssen Sie über `kms:GetPublicKey`-Berechtigungen für den KMS-Schlüssel verfügen.

```
$ 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..."
}
```

Erkennen asymmetrischer KMS-Schlüssel

Um festzustellen, ob ein bestimmter KMS-Schlüssel ein asymmetrischer KMS-Schlüssel ist, ermitteln Sie den Schlüsseltyp oder die Schlüsselspezifikation. Sie können die AWS KMS-Konsole oder die AWS KMS-API verwenden.

Einige dieser Methoden melden auch andere Aspekte der kryptografischen Konfiguration eines KMS-Schlüssels, einschließlich Schlüsselnutzung und Verschlüsselungs- oder Signaturalgorithmen, die vom KMS-Schlüssel unterstützt werden. Sie können die kryptografische Konfiguration eines KMS-Schlüssel anzeigen, jedoch nicht ändern.

Allgemeine Informationen zum Anzeigen von KMS-Schlüsseln (Sortieren, Filtern und Auswählen von Spalten für die Konsolenanzeige usw.) finden Sie unter KMS-Schlüssel in der Konsole anzeigen.

Themen

Finden des Schlüsseltyps in der KMS-Schlüsseltabelle

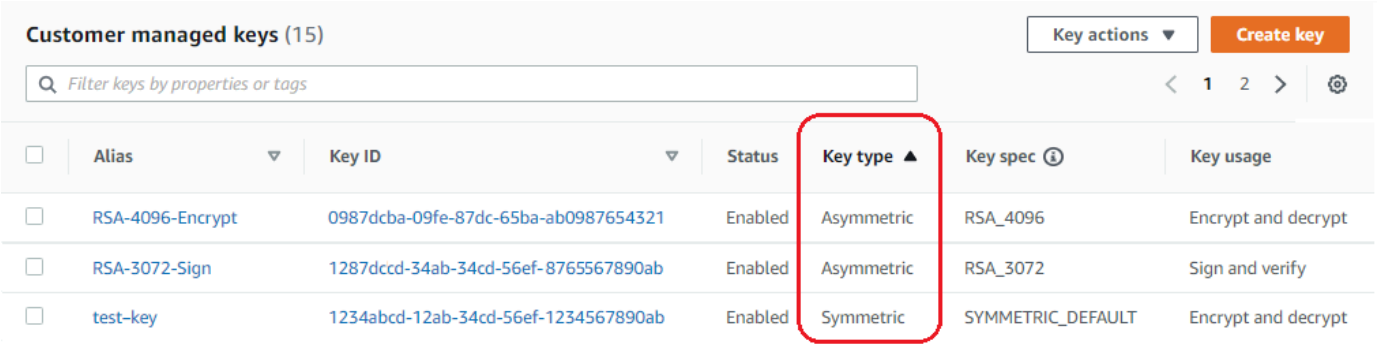
In der AWS KMS-Konsole wird in der Spalte Key type (Schlüsseltyp) angegeben, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist. Sie können eine Schlüsseltyp-Spalte zur KMS-Schlüsseltabelle auf der Seite Customer managed keys (Vom Kunden verwaltete Schlüssel) oder Von AWS verwaltete Schlüssel in der Konsole hinzufügen.

Gehen Sie folgendermaßen vor, um symmetrische und asymmetrische KMS-Schlüssel in der KMS-Schlüsseltabelle zu ermitteln.

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus.

4. In den Key type (Schlüsseltyp)-Spalte wird angegeben, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist. Sie können auch nach dem Schlüsseltyp-Wert sortieren und filtern.

   Wenn die Spalte Key type (Schlüsseltyp) nicht in der KMS-Schlüsseltabelle angezeigt wird, wählen Sie das Zahnradsymbol oben rechts auf der Seite, dann Key type (Schlüsseltyp) und schließlich Confirm (Bestätigen) aus. Sie können auch die Spalten Key spec (Schlüsselspezifikation) und Key usage (Schlüsselnutzung) hinzufügen.



Finden des Schlüsseltyps auf der Seite „Details“

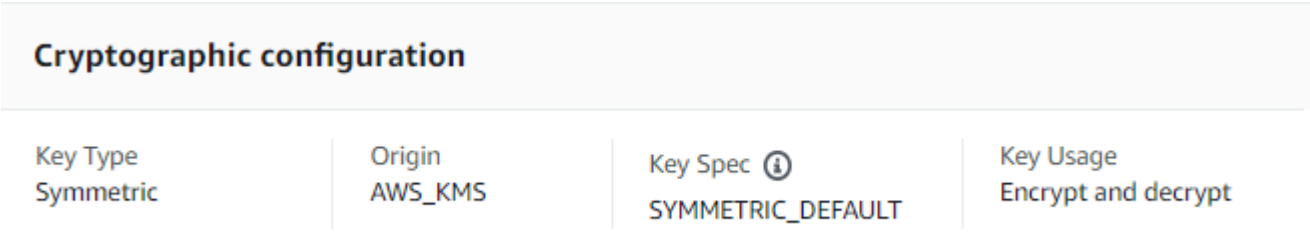
In der AWS KMS-Konsole enthält die Detailseite für jeden KMS-Schlüssel einen Bereich namens Cryptographic Configuration (kryptografische Konfiguration), der den Schlüsseltyp (symmetrisch oder asymmetrisch) und andere kryptografische Details zum KMS-Schlüssel anzeigt.

Gehen Sie folgendermaßen vor, um auf der Detailseite eines KMS-Schlüssels symmetrische und asymmetrische KMS-Schlüssel zu identifizieren.

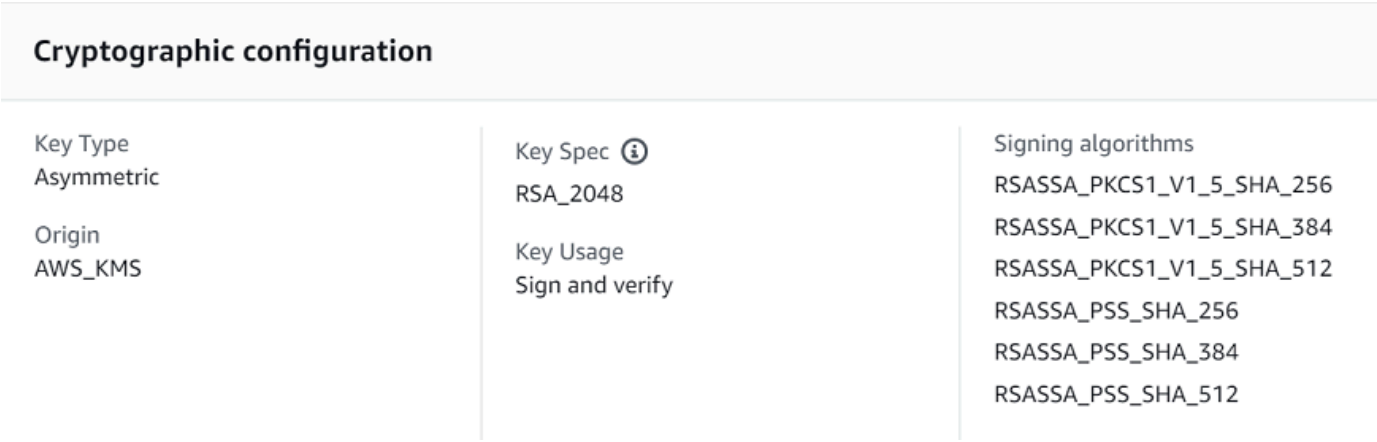
1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus. Um die Schlüssel in Ihrem Konto anzuzeigen, die AWS für Sie erstellt und verwaltet, wählen Sie im Navigationsbereich AWS managed keys (AWS-verwaltete Schlüssel) aus.
4. Wählen Sie den Alias oder die Schlüssel-ID eines KMS-Schlüssels.
5. Wählen Sie die Registerkarte Cryptographic configuration (kryptografische Konfiguration) aus. Die Registerkarte wird unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.

Die Registerkarte Cryptographic configuration (kryptografische Konfiguration) enthält den Schlüsseltyp, der angibt, ob der KMS-Schlüssel symmetrisch oder asymmetrisch ist. Außerdem werden weitere Details zum KMS-Schlüssel angezeigt, einschließlich der Schlüsselnutzung, die angibt, ob ein KMS-Schlüssel zum Verschlüsseln und Entschlüsseln oder zum Signieren und Verifizieren verwendet werden kann. Bei asymmetrischen KMS-Schlüsseln werden die vom KMS-Schlüssel unterstützten Verschlüsselungsalgorithmen oder Signaturalgorithmen angezeigt.

Das folgende ist ein Beispiel für die Registerkarte Cryptographic configuration (kryptografische Konfiguration) eines KMS-Schlüssels mit symmetrischer Verschlüsselung.



Im Folgenden finden Sie ein Beispiel für den Bereich Cryptographic configuration (kryptografische Konfiguration) für einen asymmetrischen RSA-KMS-Schlüssel, der zum Signieren und Verifizieren verwendet wird.



Finden der Schlüsselspezifikation mit der AWS KMS-API

Um festzustellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, verwenden Sie die -DescribeKeyOperation. Das `KeySpec`-Feld in der Antwort enthält die Schlüsselspezifikation des KMS-Schlüssels. Bei einem KMS-Schlüssel mit symmetrischer Verschlüsselung hat `KeySpec` den Wert `SYMMETRIC_DEFAULT`. Andere Werte weisen auf einen asymmetrischen KMS-Schlüssel oder einen HMAC-KMS-Schlüssel hin.

> **Note**
>
> Das `CustomerMasterKeySpec`-Element ist veraltet. Nutzen Sie stattdessen `KeySpec`. Um Breaking Changes zu vermeiden, enthält die `DescribeKey`-Antwort `KeySpec`- und `CustomerMasterKeySpec`-Elemente mit den gleichen Werten.

`DescribeKey` gibt für einen KMS-Schlüssel mit symmetrischer Verschlüsselung beispielsweise die folgende Antwort zurück. Der Wert von `KeySpec` ist `SYMMETRIC_DEFAULT`.

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
    "CreationDate": 1496966810.831,
    "Enabled": true,
    "Description": "",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "MultiRegion": false,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Die `DescribeKey`-Antwort für einen asymmetrischen RSA-KMS-Schlüssel, der zum Signieren und Verifizieren verwendet wird, entspricht diesem Beispiel. Der `KeySpec`-Wert ist RSA_2048 und die `KeyUsage` ist `SIGN_VERIFY`. Das `SigningAlgorithms`-Element listet die gültigen Signaturalgorithmen für den KMS-Schlüssel auf.

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1571767572.317,
    "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"
    ]
  }
}
```

Asymmetrische Schlüsselspezifikationen

Die folgenden Themen enthalten technische Informationen zu den Schlüsselspezifikationen, die AWS KMS für asymmetrische KMS-Schlüssel. Informationen zur Schlüsselspezifikation SYMMETRIC_DEFAULT für symmetrische Verschlüsselungsschlüssel sind zum Vergleich enthalten.

Themen

- RSA-Schlüsselspezifikationen
- Elliptic Curve(EC)-Schlüsselspezifikationen
- SM2-Schlüsselspezifikation (nur China-Regionen)
- Schlüsselspezifikation SYMMMETRIC_DEFAULT

RSA-Schlüsselspezifikationen

Wenn Sie eine RSA-Schlüsselspezifikation verwenden, erstellt AWS KMS einen asymmetrischen KMS-Schlüssel mit einem RSA-Schlüsselpaar. Der private Schlüssel verlässt AWS KMS niemals

unverschlüsselt. Sie können den öffentlichen Schlüssel innerhalb von AWS KMS verwenden oder für die Verwendung außerhalb von AWS KMS herunterladen.

> **Warning**
>
> Wenn Sie Daten außerhalb von AWS KMS verschlüsseln, stellen Sie sicher, dass Sie Ihren Chiffretext entschlüsseln können. Wenn Sie den öffentlichen Schlüssel aus einem KMS-Schlüssel, der aus AWS KMS gelöscht wurde, den öffentlichen Schlüssel aus einem für Signatur und Überprüfung konfigurierten KMS-Schlüssel oder einen vom KMS-Schlüssel nicht unterstützten Verschlüsselungsalgorithmus verwenden, können die Daten nicht wiederhergestellt werden.

In AWS KMS können Sie asymmetrische KMS-Schlüssel mit RSA-Schlüsselpaaren für Verschlüsselung und Entschlüsselung oder für Signatur und Verifizierung verwenden, jedoch nicht für beides. Diese Eigenschaft, die als Schlüsselnutzung bezeichnet wird, wird getrennt von der Schlüsselspezifikation bestimmt, aber Sie sollten diese Entscheidung treffen, bevor Sie eine Schlüsselspezifikation auswählen.

AWS KMS unterstützt die folgenden RSA-Schlüsselspezifikationen für Verschlüsselung und Entschlüsselung oder Signatur und Überprüfung:

- RSA_2048
- RSA_3072
- RSA_4096

Die RSA-Schlüsselspezifikationen unterscheiden sich in der Länge des RSA-Schlüssels in Bits. Die von Ihnen gewählte RSA-Schlüsselspezifikation hängt möglicherweise von Ihren Sicherheitsstandards oder den Anforderungen Ihrer Aufgabe ab. Verwenden Sie im Allgemeinen den größten Schlüssel, der für Ihre Aufgabe praktisch und erschwinglich ist. Kryptografische Operationen zu KMS-Schlüsseln mit unterschiedlichen RSA-Schlüsselspezifikationen haben unterschiedliche Preise. Weitere Informationen zur AWS KMS-Preisgestaltung finden Sie unter AWS Key Management Service – Preise. Weitere Hinweise zu Anforderungskontingenten finden Sie unter Anforderungskontingente.

RSA-Schlüsselspezifikationen für Verschlüsselung und Entschlüsselung

Wenn ein asymmetrischer RSA-KMS-Schlüssel für die Verschlüsselung und Entschlüsselung verwendet wird, verschlüsseln Sie mit dem öffentlichen Schlüssel und entschlüsseln mit dem privaten Schlüssel. Wenn Sie die `Encrypt`-Produktion in AWS KMS für einen RSA-KMS-Schlüssel aufrufen, verwendet AWS KMS den öffentlichen Schlüssel im RSA-Schlüsselpaar und den von Ihnen angegebenen Verschlüsselungsalgorithmus, um Ihre Daten zu verschlüsseln. Um den Chiffretext zu entschlüsseln, rufen Sie die `Decrypt`-Produktion auf und geben Sie denselben KMS-Schlüssel und Verschlüsselungsalgorithmus an. AWS KMS verwendet dann den privaten Schlüssel im RSA-Schlüsselpaar, um Ihre Daten zu entschlüsseln.

Sie können den öffentlichen Schlüssel auch herunterladen und ihn verwenden, um Daten außerhalb von AWS KMS zu verschlüsseln. Achten Sie darauf, einen Verschlüsselungsalgorithmus zu verwenden, den AWS KMS für RSA-KMS-Schlüssel unterstützt. Um den Chiffretext zu entschlüsseln, rufen Sie die `Decrypt`-Funktion mit demselben KMS-Schlüssel und Verschlüsselungsalgorithmus auf.

AWS KMS unterstützt zwei Verschlüsselungsalgorithmen für KMS-Schlüssel mit RSA-Schlüsselspezifikationen. Diese Algorithmen, die in PKCS #1 v2.2, definiert sind, unterscheiden sich in der Hashfunktion, die sie intern verwenden. In AWS KMS verwenden die RSAES_OAEP-Algorithmen immer dieselbe Hashfunktion sowohl für Hashzwecke als auch für die Maskengenerierungsfunktion (MGF1). Sie müssen beim Aufruf der Operationen Encrypt und Decrypt einen Verschlüsselungsalgorithmus angeben. Sie können für jede Anforderung einen anderen Algorithmus auswählen.

Unterstützte Verschlüsselungsalgorithmen für RSA-Schlüsselspezifikationen

Verschlüsselungsalgorithmus	Beschreibung des Algorithmus
RSAES_OAEP_SHA_1	PKCS #1 v2.2, Abschnitt 7.1. RSA-Versc hlüsselung mit OAEP-Auffüllung mit SHA-1 sowohl für den Hash als auch in der MGF1-Maskengenerierungsfunktion zusammen mit einer leeren Kennzeichnung.
RSAES_OAEP_SHA_256	PKCS #1, Abschnitt 7.1. RSA-Verschlüsselun g mit OAEP-Auffüllung mit SHA-256 sowohl für den Hash als auch in der MGF1-Mask

Verschlüsselungsalgorithmus	Beschreibung des Algorithmus
	engenerierungsfunktion zusammen mit einer leeren Kennzeichnung.

Sie können einen KMS-Schlüssel nicht so konfigurieren, dass er einen bestimmten Verschlüsselungsalgorithmus verwendet. Sie können jedoch die kms:EncryptionAlgorithm-Richtlinienbedingung verwenden, um die Verschlüsselungsalgorithmen anzugeben, die Prinzipale mit dem KMS-Schlüssel verwenden dürfen.

Um die Verschlüsselungsalgorithmen für einen KMS-Schlüssel abzurufen, zeigen Sie die kryptografische Konfiguration des KMS-Schlüssels in der -AWS KMSKonsole an oder verwenden Sie die -DescribeKeyOperation. stellt AWS KMS auch die Schlüsselspezifikation und die Verschlüsselungsalgorithmen bereit, wenn Sie Ihren öffentlichen Schlüssel entweder in der -AWS KMSKonsole oder mithilfe der -GetPublicKeyOperation herunterladen.

Sie können eine RSA-Schlüsselspezifikation basierend auf der Länge der Klartextdaten auswählen, die Sie in jeder Anforderung verschlüsseln können. Die folgende Tabelle zeigt die maximale Größe (in Byte) des Klartextes, den Sie bei einem einzelnen Aufruf der Produktion Encrypt verschlüsseln können. Die Werte unterscheiden sich je nach Schlüsselspezifikation und Verschlüsselungsalgorithmus. Zum Vergleich können Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung verwenden, um bis zu 4096 Bytes gleichzeitig zu verschlüsseln.

Um die maximale Klartextlänge in Bytes für diese Algorithmen zu berechnen, verwenden Sie die folgende Formel: (*Schlüsselgröße_in_Bits* /8) - (2 * *Hashlänge_in_Bits*/8) - 2. Für RSA_2048 mit SHA-256 beträgt beispielsweise die maximale Klartextgröße in Bytes (2048/8) - (2 * 256/8) -2 = 190.

Maximale Klartextgröße (in Bytes) in einer Verschlüsselungsoperation

	Verschlüsselungsalgorithmus	
Schlüsselspezifikation	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

RSA-Schlüsselspezifikationen für Signatur und Verifizierung

Wenn ein asymmetrischer RSA-KMS-Schlüssel für Signatur und Verifizierung verwendet wird, generieren Sie die Signatur für eine Nachricht mit dem privaten Schlüssel und überprüfen die Signatur mit dem öffentlichen Schlüssel.

Wenn Sie die `Sign`-Produktion in AWS KMS für einen asymmetrischen KMS-Schlüssel aufrufen, verwendet AWS KMS den privaten Schlüssel im RSA-Schlüsselpaar, die Nachricht und den von Ihnen angegebenen Signaturalgorithmus, um eine Signatur zu generieren. Um die Signatur zu überprüfen, rufen Sie die Produktion Verify auf. Geben Sie die Signatur sowie den gleichen KMS-Schlüssel, und Signaturalgorithmus und die gleiche Nachricht an. AWS KMS verwendet dann den öffentlichen Schlüssel im RSA-Schlüsselpaar, um die Signatur zu verifizieren. Sie können den öffentlichen Schlüssel auch herunterladen und ihn verwenden, um die Signatur außerhalb von AWS KMS zu überprüfen.

AWS KMS unterstützt die folgenden Signaturalgorithmen für alle KMS-Schlüssel mit einer RSA-Schlüsselspezifikation. Sie müssen einen Signaturalgorithmus angeben, wenn Sie die Operationen Sign (Signieren) und Verify (Überprüfen) aufrufen. Sie können für jede Anforderung einen anderen Algorithmus auswählen. Beim Signieren mit RSA-Schlüsselpaaren werden RSASSA-PSS-Algorithmen bevorzugt. Wir schließen RSASSA-PKCS1-v1_5-Algorithmen ein, um die Kompatibilität mit bestehenden Anwendungen zu gewährleisten.

Unterstützte Signaturalgorithmen für RSA-Schlüsselspezifikationen

Signaturalgorithmus	Beschreibung des Algorithmus
RSASSA_PSS_SHA_256	PKCS #1 v2.2, Abschnitt 8.1, RSA-Signatur mit PSS-Auffüllung mit SHA-256 sowohl für den Message Digest als auch die MGF1-Mask engenerierungsfunktion zusammen mit 256-Bit-Salt
RSASSA_PSS_SHA_384	PKCS #1 v2.2, Abschnitt 8.1, RSA-Signatur mit PSS-Auffüllung mit SHA-384 sowohl für den Message Digest als auch die MGF1-Mask engenerierungsfunktion zusammen mit 384-Bit-Salt
RSASSA_PSS_SHA_512	PKCS #1 v2.2, Abschnitt 8.1, RSA-Signatur mit PSS-Auffüllung mit SHA-512 sowohl für

Signaturalgorithmus	Beschreibung des Algorithmus
	den Message Digest als auch die MGF1-Mask engenerierungsfunktion zusammen mit 512-Bit-Salt
RSASSA_PKCS1_V1_5_SHA_256	PKCS #1 v2.2, Abschnitt 8.2, RSA-Signatur mit PKCS #1v1.5 Auffüllung und SHA-256
RSASSA_PKCS1_V1_5_SHA_384	PKCS #1 v2.2, Abschnitt 8.2, RSA-Signatur mit PKCS #1v1.5 Auffüllung und SHA-384
RSASSA_PKCS1_V1_5_SHA_512	PKCS #1 v2.2, Abschnitt 8.2, RSA-Signatur mit PKCS #1v1.5 Auffüllung und SHA-512

Sie können einen KMS-Schlüssel nicht so konfigurieren, dass bestimmte Signaturalgorithmen verwendet werden. Sie können jedoch die kms:SigningAlgorithm-Richtlinienbedingung verwenden, um die Signaturalgorithmen anzugeben, die Prinzipale mit dem KMS-Schlüssel verwenden dürfen.

Um die Signaturalgorithmen für einen KMS-Schlüssel abzurufen, zeigen Sie die kryptografische Konfiguration des KMS-Schlüssels in der -AWS KMSKonsole oder mithilfe der -DescribeKeyOperation an. stellt AWS KMS auch die Schlüsselspezifikation und Signaturalgorithmen bereit, wenn Sie Ihren öffentlichen Schlüssel entweder in der -AWS KMSKonsole oder mithilfe der -GetPublicKeyOperation herunterladen.

Elliptic Curve(EC)-Schlüsselspezifikationen

Wenn Sie eine Elliptic-Curve-Schlüsselspezifikation (ECC) verwenden, erstellt AWS KMS einen asymmetrischen KMS-Schlüssel mit einem ECC-Schlüsselpaar für Signatur und Verifizierung. Der private Schlüssel, der eine Signatur generiert, verlässt AWS KMS niemals unverschlüsselt. Sie können den öffentlichen Schlüssel verwenden, um innerhalb von AWS KMS Signaturen zu verifizieren oder für die Verwendung außerhalb von AWS KMS den öffentlichen Schlüssel herunterladen.

AWS KMS unterstützt die folgenden ECC-Schlüsselspezifikationen für asymmetrische KMS-Schlüssel.

- Asymmetrische NIST-empfohlene Elliptic Curve-Schlüsselpaare (Signatur und Verifizierung)

  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Andere asymmetrische Elliptic Curve-Schlüsselpaare (Signatur und Verifizierung)
  - ECC_SECG_P256K1 (secp256k1), häufig für Kryptowährung verwendet.

Die von Ihnen gewählte ECC-Schlüsselspezifikation hängt möglicherweise von Ihren Sicherheitsstandards oder den Anforderungen Ihrer Aufgabe ab. Verwenden Sie im Allgemeinen die Kurve mit den meisten Punkten, die für Ihre Aufgabe praktisch und erschwinglich ist.

Wenn Sie einen asymmetrischen KMS-Schlüssel für die Verwendung mit Kryptowährungen erstellen, verwenden Sie die Schlüsselspezifikation ECC_SECG_P256K1. Sie können diese Schlüsselspezifikation auch für andere Zwecke verwenden, aber sie ist für Bitcoin und andere Kryptowährungen erforderlich.

KMS-Schlüssel mit unterschiedlichen ECC-Schlüsselspezifikationen haben unterschiedliche Preise und unterliegen unterschiedlichen Anforderungkontingenten. Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service Pricing (Preise für WAF). Weitere Informationen zu Anforderungskontingenten finden Sie unter Anforderungskontingente.

Die folgende Tabelle zeigt die Signaturalgorithmen, die AWS KMS für die einzelnen ECC-Schlüsselspezifikationen unterstützt. Sie können einen KMS-Schlüssel nicht so konfigurieren, dass bestimmte Signaturalgorithmen verwendet werden. Sie können jedoch die kms:SigningAlgorithm-Richtlinienbedingung verwenden, um die Signaturalgorithmen anzugeben, die Prinzipale mit dem KMS-Schlüssel verwenden dürfen.

Unterstützte Signaturalgorithmen für ECC-Schlüsselspezifikationen

Schlüsselspezifikation	Signaturalgorithmus	Beschreibung des Algorithmus
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Abschnitt 6.4, ECDSA-Signatur unter Verwendung der durch den Schlüssel angegebenen Kurve und SHA-256 für den Message Digest.

Schlüsselspezifikation	Signaturalgorithmus	Beschreibung des Algorithmus
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, Abschnitt 6.4, ECDSA-Signatur unter Verwendung der durch den Schlüssel angegebenen Kurve und SHA-384 für den Message Digest.
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, Abschnitt 6.4, ECDSA-Signatur unter Verwendung der durch den Schlüssel angegebenen Kurve und SHA-512 für den Message Digest.
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, Abschnitt 6.4, ECDSA-Signatur unter Verwendung der durch den Schlüssel angegebenen Kurve und SHA-256 für den Message Digest.

SM2-Schlüsselspezifikation (nur China-Regionen)

Die SM2-Schlüsselspezifikation ist eine Elliptic-Curve-Schlüsselspezifikation, die in der GM/T-Spezifikationsreihe definiert ist, die von Chinas Büro für staatliche kommerzielle Kryptographie (OSCCA) veröffentlicht wurde. Die SM2-Schlüsselspezifikation ist nur in den China-Regionen verfügbar. Wenn Sie eine SM2-Schlüsselspezifikation verwenden, erstellt AWS KMS einen asymmetrischen KMS-Schlüssel mit einem SM2-Schlüsselpaar. Sie können Ihr SM2-Schlüsselpaar innerhalb von AWS KMS verwenden oder für die Verwendung außerhalb von AWS KMS herunterladen.

Im Gegensatz zur ECC-Schlüsselspezifikation können Sie einen SM2-KMS-Schlüssel zum Signieren und Überprüfen oder für die Verschlüsselung und Entschlüsselung verwenden. Sie müssen die Schlüsselverwendung bestimmen, wenn Sie den KMS-Schlüssel erstellen. Sie kann nach der Schlüsselerstellung nicht mehr geändert werden.

AWS KMS unterstützt die folgenden SM2-Verschlüsselungs- und Signaturalgorithmen:

- SM2PKE-Verschlüsselungsalgorithmus

  SM2PKE ist ein auf elliptischen Kurven basierender Verschlüsselungsalgorithmus, der von OSCCA in GM/T 0003.4-2012 definiert wurde.
- SM2DSA-Signaturalgorithmus

  SM2DSA ist ein auf elliptischen Kurven basierender Verschlüsselungsalgorithmus, der von OSCCA in GM/T 0003.2-2012 definiert wurde. SM2DSA erfordert eine unterscheidende ID, die mit dem SM3-Hashing-Algorithmus gehasht und dann mit der Nachricht oder dem Message Digest kombiniert wird, die/den Sie AWS KMS übergeben haben. Dieser verkettete Wert wird dann gehasht und von AWS KMS signiert.

Offline-Operationen mit SM2 (nur China-Regionen)

Sie können den öffentlichen Schlüssel Ihres SM2-Schlüsselpaars zur Verwendung im Offline-Betrieb herunterladen, d. h. für Operationen außerhalb von AWS KMS. Wenn Sie Ihren öffentlichen SM2-Schlüssel jedoch offline verwenden, müssen Sie möglicherweise manuell zusätzliche Konvertierungen und Berechnungen durchführen. Bei SM2DSA-Operationen müssen Sie möglicherweise eine unterscheidende ID angeben oder einen Message Digest berechnen. SM2PKE-Verschlüsselungsvorgänge erfordern möglicherweise die Konvertierung der unformatierten Geheimtext-Ausgabe in ein Format, das AWS KMS akzeptieren kann.

Um Ihnen bei diesen Vorgängen zu helfen, bietet die `SM2OfflineOperationHelper`-Klasse für Java Methoden, die die Aufgaben für Sie ausführen. Sie können diese Hilfsklasse als Modell für andere kryptografische Anbieter verwenden.

> **Important**
>
> Der `SM2OfflineOperationHelper`-Referenzcode ist so konzipiert, dass er kompatibel mit Bouncy CastleVersion 1.68 ist. Für Hilfe zu anderen Versionen wenden Sie sich an bouncycastle.org.

Offline-Überprüfung mit SM2-Schlüsselpaaren (nur China-Regionen)

Um eine Signatur außerhalb von AWS KMS mit einem öffentlichen SM2-Schlüssel zu überprüfen, müssen Sie die unterscheidende ID angeben. Wenn Sie eine unformatierte Nachricht,

`MessageType:RAW`, an die Sign-API weitergeben, verwendet AWS KMS die standardmäßige Unterscheidungs-ID, `1234567812345678`, definiert von OSCCA in GM/T 0009-2012. Sie können nicht Ihre eigene unterscheidende ID in AWS KMS angeben.

Wenn Sie jedoch einen Message Digest außerhalb von AWS generieren, können Sie Ihre eigene unterscheidende ID angeben und dann den Message Digest, `MessageType:DIGEST`, zum Signieren an AWS KMS übergeben. Dafür ändern Sie den `DEFAULT_DISTINGUISHING_ID`-Wert in der `SM2OfflineOperationHelper`-Klasse. Die von Ihnen angegebene Unterscheidungs-ID kann eine beliebige Zeichenfolge mit einer Länge von bis zu 8.192 Zeichen sein. Nachdem AWS KMS den Message Digest signiert hat, benötigen Sie entweder den Message Digest oder die Nachricht und die unterscheidende ID, die zur Berechnung des Digest verwendet wird, um ihn offline zu verifizieren.

`SM2OfflineOperationHelper`-Klasse

Innerhalb von AWS KMS werden die unformatierten Geheimtext-Konvertierungen und SM2DSA-Message-Digest-Berechnungen automatisch durchgeführt. Nicht alle kryptografischen Anbieter implementieren SM2 auf die gleiche Weise. Manche Bibliotheken, wie OpenSSL der Version 1.1.1 und höher, führen diese Aktionen automatisch aus. AWS KMS bestätigte dieses Verhalten in Tests mit OpenSSL der Version 3.0. Verwenden Sie die folgende `SM2OfflineOperationHelper`-Klasse mit Bibliotheken, wie Bouncy Castle, bei denen Sie diese Konvertierungen und Berechnungen manuell durchführen müssen.

Die `SM2OfflineOperationHelper`-Klasse bietet Methoden für die folgenden Offline-Vorgänge:

- Message-Digest-Berechnung

  Um offline einen Message Digest zu generieren, den Sie für die Offline-Überprüfung verwenden können oder den Sie zum Signieren an AWS KMS weitergeben können, verwenden Sie die `calculateSM2Digest`-Methode. Die `calculateSM2Digest`-Methode generiert einen Message Digest mit dem SM3-Hashing-Algorithmus. Die GetPublicKey API gibt Ihren öffentlichen Schlüssel im Binärformat zurück. Sie müssen den Binärschlüssel in eine Java- analysieren PublicKey. Stellen Sie den geparsten öffentlichen Schlüssel mit der Nachricht bereit. Die Methode kombiniert Ihre Nachricht automatisch mit der standardmäßigen Unterscheidungs-ID, `1234567812345678`, aber Sie können Ihre eigene Unterscheidungs-ID festlegen, indem Sie den `DEFAULT_DISTINGUISHING_ID`-Wert ändern.
- Verify

  Um eine Signatur offline zu prüfen, verwenden Sie die `offlineSM2DSAVerify`-Methode. Die `offlineSM2DSAVerify`-Methode verwendet den Message Difest, der anhand der

angegebenen unterscheidenden ID berechnet wurde, und die ursprüngliche Nachricht, die Sie zur Überprüfung der digitalen Signatur bereitstellen. Die GetPublicKey API gibt Ihren öffentlichen Schlüssel im Binärformat zurück. Sie müssen den Binärschlüssel in eine Java-analysieren PublicKey. Stellen Sie den geparsten öffentlichen Schlüssel mit der ursprünglichen Nachricht und der Signatur, die Sie überprüfen möchten, bereit. Weitere Informationen finden Sie unter .Offline-Überprüfung mit SM2-Schlüsselpaaren.

- Encrypt

  Um Klartext offline zu verschlüsseln, verwenden Sie die `offlineSM2PKEEncrypt`-Methode. Diese Methode stellt sicher, dass der Geheimtext in einem Format vorliegt, das AWS KMS entschlüsseln kann. Die `offlineSM2PKEEncrypt`-Methode verschlüsselt den Klartext und konvertiert dann den von SM2PKE erzeugten unformatierten Geheimtext in das ASN.1-Format. Die GetPublicKey API gibt Ihren öffentlichen Schlüssel im Binärformat zurück. Sie müssen den Binärschlüssel in eine Java- analysieren PublicKey. Versehen Sie den geparsten öffentlichen Schlüssel mit dem Klartext, den Sie verschlüsseln möchten.

  Wenn Sie sich nicht sicher sind, ob Sie die Konvertierung durchführen müssen, verwenden Sie den folgenden OpenSSL-Vorgang, um das Format Ihres Geheimtextes zu testen. Wenn der Vorgang fehlschlägt, müssen Sie den Geheimtext in das ASN.1-Format konvertieren.

  ```
  openssl asn1parse -inform DER -in ciphertext.der
  ```

Standardmäßig verwendet die `SM2OfflineOperationHelper`-Klasse die standardmäßige Unterscheidungs-ID, `1234567812345678`, wenn Message Digests für SM2DSA-Vorgänge generiert werden.

```
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");
    }
}
```

Schlüsselspezifikation SYMMMETRIC_DEFAULT

Die Standard-Schlüsselspezifikation SYMMMETRIC_DEFAULT ist die Schlüsselspezifikation für KMS-Schlüssel mit symmetrischer Verschlüsselung. Wenn Sie den Schlüsseltyp Symmetric (Symmetrisch) und die Schlüsselverwendung Encrypt and decrypt (Verschlüsseln und Entschlüsseln) in der AWS KMS-Konsole auswählen, wird die `SYMMETRIC_DEFAULT`-Schlüsselspezifikation ausgewählt. Wenn Sie in der -CreateKeyOperation keinen `KeySpec` Wert angeben, wird SYMMETRIC_DEFAULT ausgewählt. Wenn Sie keinen Grund haben, eine andere Schlüsselspezifikation zu verwenden, ist SYMMETRIC_DEFAULT eine gute Wahl.

SYMMETRIC_DEFAULT stellt gegenwärtig AES-256-GCM, einen symmetrischen Algorithmus, dar, der auf dem Advanced Encryption Standard (AES) im Galois Counter Mode (GCM) mit 256-Bit-Schlüsseln basiert, einem Industriestandard für sichere Verschlüsselung. Der Chiffretext, den dieser Algorithmus generiert, unterstützt zusätzliche authentifizierte Daten (AAD), z. B. einen Verschlüsselungskontext, und GCM bietet eine zusätzliche Integritätsprüfung für den Chiffretext. Details dazu finden Sie unter AWS Key Management Service kryptografische Details.

Die unter AES-256-GCM verschlüsselten Daten sind gegenwärtig und zukünftig geschützt. Kryptographen betrachten diesen Algorithmus als quantenresistent. Theoretische zukünftige, groß angelegte Quantenrechnungsangriffe auf Verschlüsselungstexte, die unter 256-Bit-AES-GCM-Schlüsseln erstellt wurden, reduzieren die effektive Sicherheit des Schlüssels auf 128-Bits. Aber diese Sicherheitsstufe reicht aus, um Brute-Force-Angriffe auf AWS KMS-Verschlüsselungstexte undurchführbar zu machen.

Die einzige Ausnahme bilden die China-Regionen, in denen SYMMETRIC_DEFAULT einen symmetrischen 128-Bit-Schlüssel darstellt, der SM4-Verschlüsselung verwendet. Sie können einen 128-Bit-SM4-Schlüssel nur innerhalb der China-Regionen erstellen. Sie können keinen 256-Bit-AES-GCM-KMS-Schlüssel in China erstellen.

Sie können einen KMS-Schlüssel mit symmetrischer Verschlüsselung in AWS KMS verwenden, um Daten zu verschlüsseln, zu entschlüsseln und erneut zu verschlüsseln und um generierte Datenschlüssel und Datenschlüsselpaare zu schützen. AWS-Services, die in AWS KMS integriert sind, verwenden im Allgemeinen KMS-Schlüssel mit symmetrischer Verschlüsselung, um Ihre Daten im Ruhezustand zu verschlüsseln. Sie können in einen KMS-Schlüssel mit symmetrischer Verschlüsselung Ihr eigenes Schlüsselmaterial importieren und KMS-Schlüssel mit symmetrischer Verschlüsselung in benutzerdefinierten Schlüsselspeichern erstellen. Eine Tabelle mit den Operationen, die Sie für symmetrische und asymmetrische KMS-Schlüssel ausführen können, finden Sie unter Vergleich symmetrischer und asymmetrischer KMS-Schlüssel.

Technische Details über AWS KMS und symmetrische Verschlüsselungsschlüssel finden Sie unter Kryptografische Details zu AWS Key Management Service.

HMAC-Schlüssel in AWS KMS

Hash-basierter Nachrichtenauthentifizierungscode (HMAC) KMS-Schlüssel sind symmetrische Schlüssel, mit denen Sie in AWS KMS HMACs generieren und überprüfen. Das eindeutige Schlüsselmaterial, das mit jedem HMAC-KMS-Schlüssel verbunden ist, liefert den geheimen Schlüssel, den HMAC-Algorithmen benötigen. Sie können einen HMAC-KMS-Schlüssel mit

`GenerateMac`- und `VerifyMac`-Operationen verwenden, um die Integrität und Authentizität von Daten in AWS KMS zu überprüfen.

HMAC-Algorithmen kombinieren eine kryptografische Hash-Funktion und einen gemeinsamen geheimen Schlüssel. Sie nehmen eine Nachricht und einen geheimen Schlüssel wie das Schlüsselmaterial in einem HMAC-KMS-Schlüssel und geben einen eindeutigen Code mit fester Größe oder Tag zurück. Wenn sich nur ein Zeichen der Nachricht ändert oder wenn der geheime Schlüssel nicht identisch ist, ist das resultierende Tag völlig anders. Durch die Notwendigkeit eines geheimen Schlüssels bietet HMAC auch Authentizität. Es ist unmöglich, ein identisches HMAC-Tag ohne den geheimen Schlüssel zu erzeugen. HMACs werden manchmal Symmetrische Signaturen genannt, weil sie wie digitale Signaturen funktionieren, aber sowohl zum Signieren als auch zur Überprüfung einen einzigen Schlüssel verwenden.

HMAC-KMS-Schlüssel und die HMAC-Algorithmen, die AWS KMS verwendet, entsprechen den Industriestandards, die in RFC 2104 festgelegt sind. Der AWS KMS GenerateMac Vorgang generiert Standard-HMAC-Tags. Diese Schlüsselpaare werden in AWS KMS-Hardwaresicherheitsmodulen generiert, die unter dem FIPS 140-2 Cryptographic Module Validation Program zertifiziert sind (außer in den Regionen China (Peking) und China (Ningxia)) und AWS KMS niemals unverschlüsselt lassen. Um einen HMAC-KMS-Schlüssel zu verwenden, müssen Sie AWS KMS aufrufen.

Sie können HMACs verwenden, um die Echtheit einer Nachricht zu ermitteln, z. B. eines JSON-Web-Token (JWT), tokenisierter Kreditkarteninformationen oder eines übermittelten Passworts. Sie können auch als sichere Key Derivation Functions (KDFs) verwendet werden, insbesondere in Anwendungen, die deterministische Schlüssel benötigen.

HMAC-KMS-Schlüssel bieten einen Vorteil gegenüber HMACs aus Anwendungssoftware, da das Schlüsselmaterial vollständig innerhalb AWS KMS generiert und verwendet wird – vorbehaltlich der Zugriffskontrollen, die Sie für den Schlüssel festgelegt haben.

> **Tip**
>
> Bewährte Methoden empfehlen, die Zeit zu beschränken, in der ein Signiermechanismus, einschließlich eines HMAC, wirksam ist. Dies schreckt einen Angriff ab, bei dem der Akteur eine signierte Nachricht verwendet, um die Gültigkeit wiederholt oder lange nach dem Ersetzen der Nachricht festzustellen. HMAC-Tags enthalten keinen Zeitstempel, aber Sie können einen Zeitstempel in das Token oder die Nachricht aufnehmen, um zu erkennen, wann es Zeit ist, den HMAC zu aktualisieren.

Autorisierte Benutzer können die HMAC-KMS-Schlüssel in Ihrem AWS-Konto erstellen, verwalten und verwenden. Dies umfasst Aktivieren und Deaktivieren von Schlüsseln, einstellen und ändern von Aliasen und Tags, und Löschen planenvon HMAC-KMS-Schlüsseln. Sie können auch IAM-Richtlinien und Erteilungen, zusammen mit Schlüsselrichtlinien, zum Steuern des Zugriffs auf Ihre HMAC-KMS-Schlüssel verwenden. Sie können alle Operationen, die Ihre HMAC-KMS-Schlüssel innerhalb von AWS nutzen oder verwalten in AWS CloudTrail-Protokollen überwachen. Sie können HMAC-KMS-Schlüssel mit importiertem Schlüsselmaterial erstellen. Sie können HMAC Multiregionale KMS-Schlüssel erstellen, die sich wie Kopien desselben HMAC-KMS-Schlüssels in mehreren AWS-Regionen verhalten.

HMAC-KMS-Schlüssel unterstützen nur die kryptografischen Operationen `GenerateMac` und `VerifyMac`. Sie können keine HMAC-KMS-Schlüssel verwenden, um Daten zu verschlüsseln oder Nachrichten zu signieren, oder eine andere Art von KMS-Schlüssel in HMAC-Operationen zu verwenden. Wenn Sie die `GenerateMac`-Operation verwenden, können Sie eine Meldung von bis zu 4 096 Bytes, einen HMAC-KMS-Schlüssel und den MAC-Algorithmus, der mit der HMAC-Schlüsselspezifikation kompatibel ist, bereitstellen und `GenerateMac` berechnet das HMAC-Tag. Um ein HMAC-Tag zu überprüfen, müssen Sie das HMAC-Tag und dieselbe Nachricht, den HMAC-KMS-Schlüssel und den MAC-Algorithmus angeben, der von `GenerateMac` verwendet wurde, um das ursprüngliche HMAC-Tag zu berechnen. Die `VerifyMac`-Operation berechnet das HMAC-Tag und überprüft, ob es mit dem bereitgestellten HMAC-Tag identisch ist. Wenn die eingegebenen und berechneten HMAC-Tags nicht identisch sind, schlägt die Überprüfung fehl.

HMAC-KMS-Schlüssel unterstützen kein(e) automatische Schlüsseldrehung und Sie können keinen HMAC-KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher erstellen.

Wenn Sie einen KMS-Schlüssel zum Verschlüsseln von Daten in einem AWS erstellen, verwenden Sie einen symmetrischen Verschlüsselungsschlüssel. Sie können keinen HMAC-KMS-Schlüssel verwenden.

Regionen

HMAC-KMS-Schlüssel werden in allen AWS-Regionen unterstützt, die AWS KMS unterstützt.

Weitere Informationen

- Hilfe bei der Auswahl eines KMS-Schlüsseltyps finden Sie unter Auswahl eines KMS-Schlüsseltyps.
- Eine Tabelle mit einem Vergleich der AWS KMS-API-Operationen, die von den einzelnen KMS-Schlüsseltypen unterstützt werden, finden Sie unter Schlüsseltypreferenz.

- Weitere Informationen zum Erstellen multiregionaler HMAC-KMS-Schlüssel finden Sie unter Schlüssel für mehrere Regionen eingeben AWS KMS.
- Informationen zum Untersuchen des Unterschieds in der Standardschlüsselrichtlinie, die die AWS KMS-Konsole für HMAC-KMS-Schlüssel festlegt, finden Sie unter the section called “Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssel mit AWS -Services”.
- Informationen zu Preisen von HMAC-KMS-Schlüsseln finden Sie unter AWS Key Management Service-Preisgestaltung.
- Weitere Informationen zu Kontingenten, die für HMAC-KMS-Schlüssel gelten, finden Sie unter Ressourcenkontingente und Anforderungskontingente.
- Weitere Informationen zum Löschen von HMAC-KMS-Schlüsseln finden Sie unter Löschen von AWS KMS keys.
- Weitere Informationen über die Verwendung von HMACs zur Erstellung von JSON-Web-Tokens finden Sie unter So schützen Sie HMACs in AWS KMS im AWS -Sicherheits-Blog.
- Hören Sie sich einen Podcast an: Einführung von HMACs für AWS Key Management Service im The Official AWS Podcast.

Themen

Schlüsselspezifikationen für HMAC-KMS-Schlüssel

AWS KMS unterstützt symmetrische HMAC-Schlüssel in unterschiedlichen Längen. Häufig wird die von Ihnen ausgewählte Schlüsselspezifikation durch Ihre Sicherheits-, gesetzliche, oder geschäftliche Anforderungen bestimmt. Die Länge des Schlüssels bestimmt den MAC-Algorithmus, der in den VerifyMac Operationen GenerateMac und verwendet wird. Im Allgemeinen sind längere Schlüssel sicherer. Verwenden Sie den längsten Schlüssel, der für Ihren Anwendungsfall praktisch ist.

HMAC-Schlüsselspezifikation	MAC-Algorithmen
HMAC_224	HMAC_SHA_224

HMAC-Schlüsselspezifikation	MAC-Algorithmen
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Erstellen von HMAC-KMS-Schlüsseln

Sie können HMAC-KMS-Schlüssel in der AWS KMS-Konsole erstellen, indem Sie die CreateKey-API oder eine AWS CloudFormation-Vorlage verwenden.

AWS KMS unterstützt mehrere Schlüsselspezifikationen für HMAC-KMS-Schlüssel. Die von Ihnen ausgewählte Schlüsselspezifikation wird möglicherweise durch gesetzliche, Sicherheits- oder geschäftliche Anforderungen bestimmt. Im Allgemeinen sind längere Schlüssel Brute-Force-Angriffen gegenüber weniger anfällig.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Wenn Sie einen KMS-Schlüssel erstellen, um Daten zu verschlüsseln, die Sie in einem AWS-Service speichern oder verwalten, verwenden Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung. AWS-Services, die mit AWS KMS integriert werden, unterstützen keine asymmetrischen KMS-Schlüssel oder HMAC-KMS-Schlüssel. Hilfe zum Erstellen eines KMS-Schlüssels mit symmetrischer Verschlüsselung finden Sie unter Erstellen von Schlüsseln.

Weitere Informationen

- Um festzustellen, welche Art von KMS-Schlüssel erstellt werden soll, beziehen Sie sich auf Auswahl eines KMS-Schlüsseltyps.
- Sie können die in diesem Thema beschriebenen Verfahren verwenden, um einen multiregionalen primären HMAC-KMS-Schlüssel zu erstellen. Um einen multiregionalen HMAC-Schlüssel zu replizieren, beziehen Sie sich auf the section called “Erstellen von Replikatschlüsseln”.

- Weitere Informationen über die Berechtigungen, die zum Erstellen von KMS-Schlüsseln erforderlich sind, finden Sie unter Berechtigungen zum Erstellen von KMS-Schlüsseln.
- Informationen zur Verwendung einer -AWS CloudFormationVorlage zum Erstellen eines HMAC-KMS-Schlüssels finden Sie unter AWS::KMS::Key im AWS CloudFormation -Benutzerhandbuch.

Themen

Erstellen von HMAC-KMS-Schlüsseln (Konsole)

Sie können die AWS Management Console zum Erstellen von HMAC-KMS-Schlüsseln verwenden. HMAC-KMS-Schlüssel sind symmetrische Schlüssel mit einer Schlüsselverwendung von Generate and verify MAC (MAC generieren und überprüfen). Sie können auch multiregionale HMAC-KMS-Schlüssel erstellen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Wählen Sie für Key type (Schlüsseltyp) Symmetric (Symmetrisch).

   HMAC-KMS-Schlüssel sind symmetrisch. Sie verwenden denselben Schlüssel, um HMAC-Tags zu generieren und zu überprüfen.
6. Wählen Sie für Schlüsselverwendung Generate and verify MAC (MAC Generieren und überprüfen) aus.

   MAC generieren und überprüfen ist die einzig gültige Schlüsselverwendung für HMAC-KMS-Schlüssel.

> **Note**
>
> Key usage (Schlüsselverwendung) wird für symmetrische Schlüssel nur angezeigt, wenn HMAC-KMS-Schlüssel in Ihrer ausgewählten Region unterstützt werden.

7. Wählen Sie eine Spezifikation (Key spec (Schlüsselspezifikation)) für Ihren HMAC-KMS-Schlüssel aus.

   Die von Ihnen ausgewählte Schlüsselspezifikation kann möglicherweise durch gesetzliche, Sicherheits- oder geschäftliche Anforderungen bestimmt werden. Im Allgemeinen sind längere Schlüssel sicherer.

8. Um einen multiregionalen primären HMAC-Schlüssel zu erstellen, wählen Sie unter Advanced Options (Advanced Optionen) Multi-Region key (Multiregionaler Schlüssel) aus. Die freigegebenen Eigenschaften, die Sie für diesen KMS-Schlüssel definieren, wie z. B. den Schlüsseltyp und die Schlüsselverwendung, wird mit seinen Replikatschlüsseln geteilt. Details hierzu finden Sie unter Erstellen von multiregionalen Schlüsseln.

   Sie können dieses Verfahren nicht verwenden, um einen Replikatschlüssel zu erstellen. Um einen multiregionalen Replikat-HMAC-Schlüssel, befolgen Sie die Anweisungen zum Erstellen eines Replikatschlüssels.

9. Wählen Sie Weiter aus.

10. Geben Sie einen Alias für den KMS-Schlüssel ein. Der Aliasname darf nicht mit **`aws/`** beginnen. Das Präfix **`aws/`** ist von Amazon Web Services reserviert und steht für Von AWS verwaltete Schlüssel in Ihrem Konto.

    Wir empfehlen Ihnen, einen Alias zu verwenden, der den KMS-Schlüssel als HMAC-Schlüssel identifiziert, z. B. `HMAC/test-key`. Dies erleichtert die Identifizierung Ihrer HMAC-Schlüssel in der AWS KMS-Konsole, in der Sie Schlüssel nach Tags und Aliasen sortieren und filtern können, jedoch nicht nach Schlüsselspezifikation oder Schlüsselverwendung.

    Zum Erstellen eines KMS-Schlüssels in der Konsole benötigen Sie Aliase AWS Management Console. Sie können keinen Alias angeben, wenn Sie die -CreateKeyOperation verwenden, aber Sie können die Konsole oder die -CreateAliasOperation verwenden, um einen Alias für einen vorhandenen KMS-Schlüssel zu erstellen. Details hierzu finden Sie unter Verwenden von Aliassen.

11. (Optional) Geben Sie eine Beschreibung für den KMS-Schlüssel ein.

Geben Sie eine Beschreibung ein, die die Art von Daten, die Sie schützen möchten, oder die Anwendung, die Sie mit dem KMS-Schlüssel verwenden möchten, erklärt.

Sie können jetzt eine Beschreibung hinzufügen oder sie jederzeit aktualisieren, es sei denn, der Schlüsselstatus lautet `Pending Deletion` oder `Pending Replica Deletion`. Um die Beschreibung eines vorhandenen kundenverwalteten Schlüssels hinzuzufügen, zu ändern oder zu löschen, bearbeiten Sie die Beschreibung in der AWS Management Console oder verwenden Sie die -UpdateKeyDescriptionOperation.

12. (Optional) Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wählen Sie Add tag (Tag hinzufügen), wenn Sie mehr als ein Tag zum KMS-Schlüssel hinzufügen möchten.

    Erwägen Sie, ein Tag hinzuzufügen, das den Schlüssel als HMAC-Schlüssel identifiziert, z. B. `Type=HMAC`. Dies erleichtert die Identifizierung Ihrer HMAC-Schlüssel in der AWS KMS-Konsole, in der Sie Schlüssel nach Tags und Aliasen sortieren und filtern können, jedoch nicht nach Schlüsselspezifikation oder Schlüsselverwendung.

    Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

13. Wählen Sie Weiter aus.

14. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel verwalten können.

    > **Note**
    >
    > Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung zum Verwalten des KMS-Schlüssels erteilen. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.
    > Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

15. (Optional) Um zu verhindern, dass die ausgewählten IAM-Benutzer und -Rollen diesen KMS-Schlüssel löschen, deaktivieren Sie unten auf der Seite im Abschnitt Key deletion

(Schlüssellöschung) das Kontrollkästchen Allow key administrators to delete this key (Administratoren erlauben, diesen Schlüssel zu löschen).

16. Wählen Sie Weiter aus.
17. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel für kryptographische Operationen verwenden können.

> **Note**
>
> Diese wichtige Richtlinie gibt AWS-Konto volle Kontrolle über diesen KMS-Schlüssel. Kontoadministratoren können damit anderen Prinzipalen mithilfe von IAM-Richtlinien die Berechtigung erteilen, den KMS-Schlüssel in kryptografischen Operationen zu verwenden. Details hierzu finden Sie unter the section called “Standardschlüsselrichtlinie”.
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

18. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other AWS-Konten (Andere Konten) unten auf der Seite die Option Add another AWS-Konto (Weiteres Konto hinzufügen) und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

> **Note**
>
> Um auch Prinzipalen aus den externen Konten Zugriff auf den KMS-Schlüssel zu erlauben, müssen die Administratoren der externen Konten IAM-Richtlinien erstellen, die diese Berechtigungen bereitstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

19. Wählen Sie Weiter.
20. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
21. Wählen Sie Finish (fertigstellen) aus, um den HMAC-KMS-Schlüssel zu erstellen.

Erstellen von HMAC-KMS-Schlüsseln (AWS KMS-API)

Sie können die -CreateKeyOperation verwenden, um einen HMAC-KMS-Schlüssel zu erstellen. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Wenn Sie einen HMAC-KMS-Schlüssel erstellen, müssen Sie den `KeySpec`-Parameter angeben, der den Typ der von Ihnen erstellten KMS-Schlüssel bestimmt. Außerdem müssen Sie einen `KeyUsage`-Wert von GENERATE_VERIFY_MAC angeben, obwohl es der einzig gültige Wert für die Schlüsselverwendung für HMAC-Schlüssel ist. Um einen multiregionalen HMAC-KMS-Schlüssel zu erstellen, fügen Sie den `MultiRegion`-Parameter mit einem Wert von `true` hinzu. Diese Eigenschaften können nicht geändert werden, nachdem der KMS-Schlüssel erstellt wurde.

Mit der -`CreateKey`Operation können Sie keinen Alias angeben, aber Sie können die -CreateAliasOperation verwenden, um einen Alias für Ihren neuen KMS-Schlüssel zu erstellen. Wir empfehlen Ihnen, einen Alias zu verwenden, der den KMS-Schlüssel als HMAC-Schlüssel identifiziert, z. B. `HMAC/test-key`. Dies erleichtert die Identifizierung Ihrer HMAC-Schlüssel in der AWS KMS-Konsole, in der Sie Schlüssel nach Alias sortieren und filtern können, jedoch nicht nach Schlüsselspezifikation oder Schlüsselverwendung.

Wenn Sie versuchen, einen HMAC-KMS-Schlüssel in einer AWS-Region zu erstellen, in der HMAC-Schlüssel nicht unterstützt werden, gibt die `CreateKey`-Operation eine `UnsupportedOperationException` zurück.

Im folgenden Beispiel wird die `CreateKey`-Operation zum Erstellen eines 512-Bit-HMAC-KMS-Schlüssels verwendet.

```
$ 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
    }
}
```

Steuern des Zugriffs auf HMAC-KMS-Schlüssel

Um den Zugriff auf einen HMAC-KMS-Schlüssel zu steuern, verwenden Sie eine Schlüsselrichtlinie, die für jeden KMS-Schlüssel erforderlich ist. Sie können auch IAM-Richtlinien und Erteilungen verwenden.

Die Standardschlüsselrichtlinie für HMAC-Schlüssel, die in der AWS KMS-Konsole erstellt wird, gibt wichtigen Benutzern die Berechtigung zum Aufrufen der `GenerateMac`- und `VerifyMac`-Operationen. Es enthält jedoch nicht die Schlüsselrichtlinie entwickelt für die Verwendung von Erteilungen mit AWS-Services. Wenn Sie HMAC-Schlüssel mit dem `CreateKey`-Operation erstellen, müssen Sie diese Berechtigungen in der Schlüsselrichtlinie oder einer IAM-Richtlinie angeben.

Sie können Globale Bedingungsschlüssel von AWS und AWS KMS-Bedingungsschlüssel zum Verfeinern und Beschränken von Berechtigungen auf HMAC-Schlüssel verwenden. Sie können beispielsweise die `kms:ResourceAliases`-Bedingungsschlüssel zur Steuerung des Zugriffs auf AWS KMS-Operationen basierend auf den Aliasen, die mit einem HMAC-Schlüssel verknüpft sind, verwenden. Folgende AWS KMS-Richtlinienbedingungen sind nützlich für Richtlinien zu HMAC-Schlüsseln.

- Verwenden Sie einen `kms:MacAlgorithm`-Bedingungsschlüssel, um die Algorithmen zu begrenzen, die die Prinzipale anfordern können, wenn sie `GenerateMac`- und `VerifyMac`-Operationen aufrufen. Sie können beispielsweise zulassen, dass Prinzipale die `GenerateMac`-Operationen aufrufen, aber nur wenn der MAC-Algorithmus in der Anforderung `HMAC_SHA_384` ist.
- Verwenden Sie einen `kms:KeySpec`-Bedingungsschlüssel, um zu erlauben oder zu verhindern, dass Prinzipale bestimmte Arten von HMAC-Schlüsseln erstellen. Um beispielsweise Prinzipalen zu erlauben, nur HMAC-Schlüssel zu erstellen, können Sie die -CreateKeyOperation zulassen, aber die -`kms:KeySpec`Bedingung verwenden, um nur Schlüssel mit einer -`HMAC_384`Schlüsselspezifikation zuzulassen.

Sie können auch den `kms:KeySpec`-Bedingungsschlüssel verwenden, um den Zugriff auf andere Operationen auf einen KMS-Schlüssel basierend auf der Schlüsseleigenschaft des Schlüssels verwenden. Sie können beispielsweise zulassen, dass Prinzipale das Löschen von Schlüsseln nur für KMS-Schlüssel mit einer `HMAC_256`-Schlüsselspezifikation planen und stornieren.

- Verwenden Sie einen `kms:KeyUsage`-Bedingungsschlüssel, um zu erlauben oder zu verhindern, dass Prinzipale jegliche Arten von HMAC-Schlüsseln erstellen. Um beispielsweise Prinzipalen zu erlauben, nur HMAC-Schlüssel zu erstellen, können Sie die -CreateKeyOperation zulassen, aber die -`kms:KeyUsage`Bedingung verwenden, um nur Schlüssel mit einer `GENERATE_VERIFY_MAC` Schlüsselnutzung zuzulassen.

  Sie können auch den `kms:KeyUsage`-Bedingungsschlüssel verwenden, um den Zugriff auf andere Operationen auf einen KMS-Schlüssel basierend auf der Schlüsseleigenschaft des Schlüssels zu steuern. Sie können beispielsweise Prinzipalen nur die Aktivierung und Deaktivierung von KMS-Schlüsseln mit einer `GENERATE_VERIFY_MAC`-Schlüsselverwendung erlauben.

Sie können auch Erteilungen für `GenerateMac`- und `VerifyMac`-Operationen, die Erteilungsoperationen sind, erstellen. In einer Erteilung für einen HMAC-Schlüssel können Sie jedoch keine Verschlüsselungskontext-Erteilungs-Einschränkungen verwenden. Das HMAC-Tag-Format unterstützt keine Verschlüsselungskontextwerte.

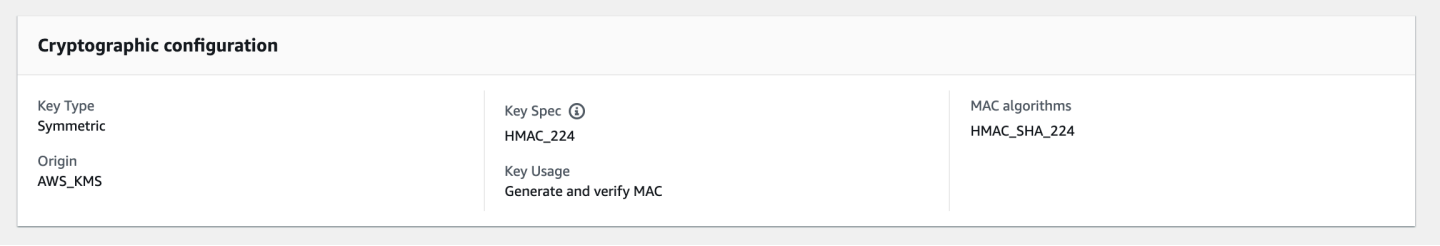
Anzeigen von HMAC-KMS-Schlüsseln

Sie können HMAC-KMS-Schlüssel in der AWS KMS-Konsole oder über die DescribeKey-API anzeigen. Sie können die Verwendung Ihrer HMAC-KMS-Schlüssel in -AWS CloudTrailProtokollen und in Amazon CloudWatch überwachen. Grundlegende Anweisungen zum Anzeigen von KMS-Schlüsseln finden Sie unter Anzeigen von Schlüsseln.

Sie können HMAC-KMS-Schlüssel von anderen KMS-Schlüsseltypen durch ihre Schlüsselspezifikation, die mit `HMAC` beginnen, oder deren Schlüsselverwendung, die immer Generate and verify MAC (MAC generieren und überprüfen) (`GENERATE_VERIFY_MAC`) ist, unterscheiden.

HMAC-KMS-Schlüssel sind in der Tabelle auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) der AWS KMS-Konsole enthalten. Sie können jedoch KMS-Schlüssel nicht nach Schlüsselspezifikation oder Schlüsselverwendung sortieren oder filtern. Um das Auffinden Ihrer HMAC-Schlüssel zu erleichtern, weisen Sie ihnen einen unverwechselbaren Alias oder Tag zu. Dann können Sie nach dem Alias oder Tag sortieren oder filtern.

Auf der Seite „Schlüsseldetails“ für einen HMAC-KMS-Schlüssel finden Sie seine Konfigurationsdetails auf der Registerkarte Cryptographic configuration (Kryptografische Konfiguration).



Schlüssel für mehrere Regionen eingeben AWS KMS

AWS KMS unterstützt Schlüssel für mehrere Regionen, AWS-Regionen die unterschiedlich sind und synonym verwendet werden können — als ob Sie denselben Schlüssel AWS KMS keys in mehreren Regionen hätten. Jeder Satz verwandter Schlüssel für mehrere Regionen hat dasselbe Schlüsselmaterial und dieselbe Schlüssel-ID, sodass Sie Daten in einem Schlüssel verschlüsseln AWS-Region und in einem anderen entschlüsseln können, AWS-Region ohne sie erneut zu verschlüsseln oder regionsübergreifend aufzurufen. AWS KMS

Wie alle KMS-Schlüssel bleiben Schlüssel für mehrere Regionen niemals unverschlüsselt. AWS KMS Sie können symmetrische oder asymmetrische Schlüssel mit mehreren Regionen für die Verschlüsselung oder Signierung erstellen, HMAC-Schlüssel für mehrere Regionen zum Generieren und Überprüfen von HMAC-Tags und Schlüssel mit mehreren Regionen mit importiertem Schlüsselmaterial oder generiertem Schlüsselmaterial erstellen. AWS KMS Sie müssen jeden multiregionalen Schlüssel verwalten, unabhängig voneinander, einschließlich der Erstellung von Aliasen und Tags und der Festlegung ihrer wichtigsten Richtlinien und Berechtigungen sowie der selektiven Aktivierung und Deaktivierung. Sie können multiregionale Schlüssel in allen kryptografischen Operationen verwenden, die Sie mit einzelregionalen Schlüsseln ausführen können.

Multiregionale Schlüssel sind eine flexible und leistungsstarke Lösung für viele gängige Datensicherheitsszenarien.

Notfallwiederherstellung

> In einer Sicherungs- und Wiederherstellungsarchitektur können Sie mit Schlüsseln für mehrere Regionen verschlüsselte Daten auch bei einem Ausfall unterbrechungsfrei verarbeiten. AWS-Region Daten, die in Backup-Regionen verwaltet werden, können im Backup-Bereich entschlüsselt werden, und Daten, die im Backup-Bereich neu verschlüsselt wurden, können in der primären Region entschlüsselt werden, wenn diese Region wiederhergestellt wird.

Globales Datenmanagement

Unternehmen, die global tätig sind, benötigen global verteilte Daten, die konsistent über AWS-Regionen verfügbar sind. Sie können multiregionale Schlüssel in allen Regionen erstellen, in denen sich Ihre Daten befinden. Anschließend können Sie die Schlüssel so verwenden, als wären sie einzelregionale Schlüssel, ohne die Latenz eines regionsübergreifenden Anrufs oder die Kosten für die erneute Verschlüsselung von Daten unter einem anderen Schlüssel in jeder Region.

Verteilte Signaturanwendungen

Anwendungen, die regionsübergreifende Signaturfunktionen erfordern, können asymmetrische Signaturschlüssel für mehrere Regionen verwenden, um identische digitale Signaturen konsistent und wiederholt in verschiedenen AWS-Regionen zu generieren.

Wenn Sie die Zertifikatverkettung mit einem einzigen globalen Vertrauensspeicher (für eine einzelne Stammzertifizierungsstelle) und regionale Zwischenzertifizierungsstellen verwenden, die von der Stammzertifizierungsstelle signiert sind, benötigen Sie keine multiregionalen Schlüssel. Wenn Ihr System jedoch keine Zwischenzertifizierungsstellen unterstützt, z. B. Anwendungssignierung, können Sie multiregionale Schlüssel verwenden, um die Konsistenz der regionalen Zertifizierungen zu gewährleisten.

Aktiv/aktiv-Anwendungen, die sich über mehrere Regionen erstrecken

Einige Arbeitslasten und Anwendungen können sich über mehrere Regionen in aktiv/aktiv-Architekturen erstrecken. Für diese Anwendungen können multiregionale Schlüssel die Komplexität reduzieren, indem sie dasselbe Schlüsselmaterial für gleichzeitige Verschlüsselungs- und Entschlüsselungsoperationen für Daten bereitstellen, die möglicherweise über Regionsgrenzen hinweg verschoben werden.

Sie können multiregionale Schlüssel mit clientseitigen Verschlüsselungsbibliotheken verwenden, z. B. die AWS Encryption SDK, den DynamoDB Encryption Client und clientseitige Simple Storage Service (Amazon S3)-Verschlüsselung. Ein Beispiel für die Verwendung von Regionsschlüsseln mit globalen Amazon DynamoDB-Tabellen und dem DynamoDB Encryption Client finden Sie im Security Blog unter Clientseitige Verschlüsselung globaler Daten mit Schlüsseln für mehrere Regionen. AWS KMS AWS

AWS Dienste, die AWS KMS für Verschlüsselung im Ruhezustand oder digitale Signaturen integriert sind, behandeln Schlüssel mit mehreren Regionen derzeit so, als ob es sich um Schlüssel mit nur einer Region handeln würde. Sie können Daten, die zwischen Regionen verschoben wurden,

erneut verpacken oder neu verschlüsseln. Beispielsweise entschlüsselt und verschlüsselt die regionsübergreifende Replikation von Simple Storage Service (Amazon S3) Daten unter einem KMS-Schlüssel in der Zielregion, selbst wenn Objekte repliziert werden, die durch einen multiregionalen Schlüssel geschützt sind.

Multiregionale Schlüssel sind nicht global. Sie erstellen einen multiregionalen Primärschlüssel und replizieren ihn dann in Regionen, die Sie in einer AWS -Partition auswählen. Anschließend verwalten Sie den multiregionalen Schlüssel in jeder Region unabhängig. AWS Weder erstellt noch AWS KMS jemals automatisch Schlüssel für mehrere Regionen in Ihrem Namen oder repliziert sie in eine Region. Von AWS verwaltete Schlüssel, bei den KMS-Schlüsseln, die AWS Dienste in Ihrem Konto für Sie erstellen, handelt es sich immer um Schlüssel für einzelne Regionen.

Sie können einen vorhandenen einzelregionalen Schlüssel nicht in einen multiregionalen Schlüssel konvertieren. Dieses Merkmal stellt sicher, dass alle Daten, die mit vorhandenen einzelregionalen Schlüsseln geschützt sind, dieselben Datenresidenz- und Datensouveränitäts-Eigenschaften beibehalten.

Für die meisten Datensicherheitsanforderungen sind Standardschlüssel für AWS KMS einzelne Regionen aufgrund der regionalen Isolierung und Fehlertoleranz regionaler Ressourcen die am besten geeignete Lösung. Wenn Sie jedoch Daten in clientseitigen Anwendungen über mehrere Regionen hinweg verschlüsseln oder signieren müssen, sind multiregionale Schlüssel möglicherweise die Lösung.

Regionen

Schlüssel für mehrere Regionen werden in allen Ländern AWS KMS unterstützt AWS-Regionen , mit Ausnahme von China (Peking) und China (Ningxia).

Preise und Kontingente

Jeder Schlüssel in einem Satz verwandter multiregionaler Schlüssel zählt als ein KMS-Schlüssel für Preise und Kontingente. AWS KMS -Kontingente werden separat für jede Region eines Kontos berechnet. Die Verwendung und Verwaltung der multiregionalen Schlüssel in jeder Region zählen zu den Kontingenten für diese Region.

Unterstützte KMS-Schlüsseltypen

Sie können die folgenden Typen von für mehrere Regionen geltenden KMS-Schlüsseln erstellen:

- KMS-Schlüssel zur symmetrischen Verschlüsselung
- Asymmetrische KMS-Schlüssel
- HMAC-KMS-Schlüssel
- KMS-Schlüssel mit importiertem Schlüsselmaterial

Sie können keine multiregionale Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden.

Themen

Sicherheitsaspekte für multiregionale Schlüssel

Verwenden Sie einen Schlüssel für AWS KMS mehrere Regionen nur, wenn Sie einen benötigen. Multiregionale Schlüssel bieten eine flexible und skalierbare Lösung für Workloads, die verschlüsselte Daten zwischen AWS-Regionen verschieben oder regionsübergreifenden Zugang benötigen. Verwenden Sie einen multiregionalen Schlüssel, wenn Sie geschützte Daten über Regionen hinweg freigeben, verschieben oder sichern müssen oder identische digitale Signaturen von Anwendungen erstellen müssen, die in verschiedenen Regionen ausgeführt werden.

Beim Erstellen eines multiregionalen Schlüssels wird das Schlüsselmaterial jedoch innerhalb AWS KMSüber die Grenzen der AWS-Region verschoben. Der Chiffretext, der von einem multiregionalen Schlüssel generiert wird, kann möglicherweise von mehreren verwandten Schlüsseln an mehreren geografischen Standorten entschlüsselt werden. Regional isolierte Services und Ressourcen bieten auch erhebliche Vorteile. Jede AWS-Region ist isoliert und unabhängig von den anderen Regionen. Regionen bieten Fehlertoleranz, Stabilität und Ausfallsicherheit und können auch die Latenz verkürzen. Sie ermöglichen das Erstellen redundanter Ressourcen, die verfügbar bleiben und von einem Ausfall in einer anderen Region nicht betroffen sind. Außerdem stellen sie sicher AWS KMS, dass jeder Chiffretext mit nur einem Schlüssel entschlüsselt werden kann.

Multiregionale Schlüssel werfen auch neue Sicherheitsbedenken auf.

- Die Steuerung des Zugriffs und die Durchsetzung von Datensicherheitsrichtlinien ist mit multiregionalen Schlüsseln komplexer. Sie müssen sicherstellen, dass die Richtlinie bei Schlüsseln in mehreren isolierten Regionen konsistent überwacht wird. Und Sie müssen Richtlinien verwenden, um Grenzen zu erzwingen, anstatt sich auf separate Schlüssel zu verlassen.

  Sie müssen beispielsweise Richtlinienbedingungen für Daten festlegen, um zu verhindern, dass Lohn- und Gehaltsabrechnungsteams in einer Region Lohn- und Gehaltsabrechnungsdaten für eine andere Region lesen können. Außerdem müssen Sie die Zugriffssteuerung verwenden, um ein Szenario zu verhindern, in dem ein multiregionaler Schlüssel in einer Region die Daten eines Mandanten schützt und ein verwandter multiregionaler Schlüssel in einer anderen Region die Daten eines anderen Mandanten schützt.
- Die Prüfung von Schlüsseln über Regionen hinweg ist ebenfalls komplexer. Mit multiregionalen Schlüsseln müssen Sie Prüfungsaktivitäten über mehrere Regionen hinweg überprüfen und abstimmen, um ein vollständiges Verständnis der wichtigsten Aktivitäten für geschützte Daten zu erhalten.
- Die Einhaltung von Datenresidenz-Mandaten kann komplexer sein. Mit isolierten Regionen können Sie die Einhaltung von Datenresidenz und Datenhoheit sicherstellen. KMS-Schlüssel in einer bestimmten Region können sensible Daten nur in dieser Region entschlüsseln. Daten, die in einer Region verschlüsselt sind, können in jeder anderen Region vollständig geschützt und unzugänglich bleiben.

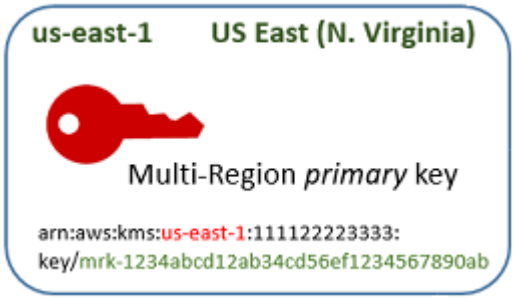
  Um die Datenresidenz und Datensouveränität mit Schlüsseln für mehrere Regionen zu überprüfen, müssen Sie Zugriffsrichtlinien implementieren und Ereignisse für mehrere Regionen zusammenstellen AWS CloudTrail .

Um Ihnen die Verwaltung der Zugriffskontrolle für Schlüssel mit mehreren Regionen zu erleichtern, ist die Berechtigung zum Replizieren eines Schlüssels für mehrere Regionen (kms: ReplicateKey) von der Standardberechtigung zum Erstellen von Schlüsseln (kms:) getrennt. CreateKey AWS KMS Unterstützt außerdem mehrere Richtlinienbedingungen für Schlüssel mit mehreren Regionen, z. B. die Erlaubnis`kms:MultiRegion`, Schlüssel für mehrere Regionen zu erstellen, zu verwenden oder zu verwalten, oder die die Regionen einschränkt`kms:ReplicaRegion`, in die ein Schlüssel für mehrere Regionen repliziert werden kann. Details hierzu finden Sie unter Steuern des Zugriffs auf multiregionale Schlüssel.

Funktionsweise von multiregionalen Schlüsseln

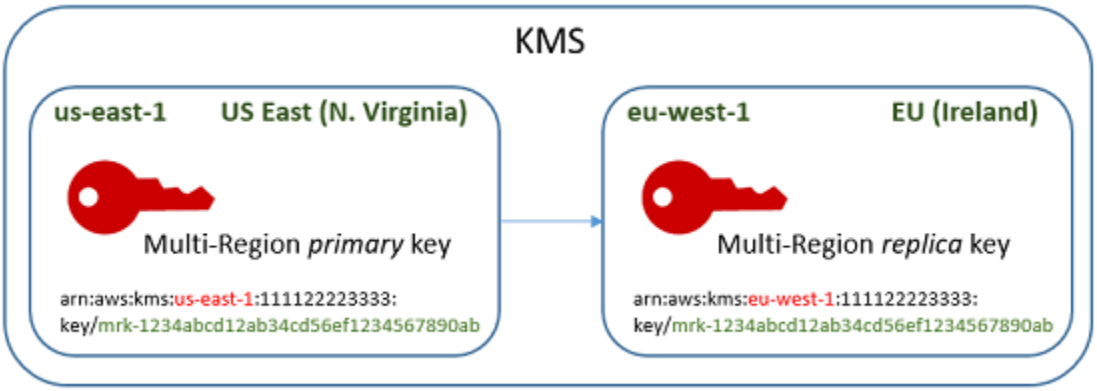
Sie beginnen mit der Erstellung eines symmetrischen oder asymmetrischen Primärschlüssels für mehrere Regionen in einem System, AWS-Region das dies AWS KMS unterstützt, z. B. US East (Nord-Virginia). Sie entscheiden nur, wenn Sie ihn erstellen, ob es sich bei einem Schlüssel um einen einzelregionalen Schlüssel oder um einen multiregionalen Schlüssel handelt. Sie können diese Eigenschaft später nicht ändern. Wie bei jedem KMS-Schlüssel, legen Sie eine Schlüsselrichtlinie für den multiregionalen Schlüssel fest, und Sie können Berechtigungen erstellen und Aliase und Tags für die Kategorisierung und Autorisierung hinzufügen. (Diese sind unabhängige Eigenschaften, die nicht gemeinsam genutzt oder mit anderen Schlüsseln synchronisiert werden.) Sie können Ihren multiregionalen Primärschlüssel in kryptografischen Operationen zur Verschlüsselung oder Signatur verwenden.

Sie können einen Primärschlüssel für mehrere Regionen in der AWS KMS Konsole oder mithilfe der CreateKeyAPI erstellen, wobei der Parameter auf gesetzt ist`MultiRegion`. `true` Beachten Sie, dass multiregionale Schlüssel eine unverwechselbare Schlüssel-ID haben, die mit `mrk-` beginnt. Sie können das `mrk-`-Präfix verwenden, um multiregionale Schlüssel programmgesteuert zu identifizieren.



Wenn Sie möchten, können Sie den Primärschlüssel für mehrere Regionen in einen oder mehrere Primärschlüssel in derselben AWS Partition replizieren, z. B. AWS-Regionen in Europa (Irland). Wenn Sie dies tun, AWS KMS wird in der angegebenen Region ein Replikatschlüssel mit derselben Schlüssel-ID und anderen gemeinsamen Eigenschaften wie der Primärschlüssel erstellt. Dann transportiert es das Schlüsselmaterial sicher über die Regionsgrenze und ordnet es dem neuen KMS-Schlüssel in der Zielregion zu, alles innerhalb von AWS KMS. Das Ergebnis sind zwei verwandte multiregionale Schlüssel – ein Primärschlüssel und ein Replikatschlüssel – die austauschbar verwendet werden können.

Sie können einen Replikatschlüssel für mehrere Regionen in der AWS KMS Konsole oder mithilfe der API erstellen. ReplicateKey



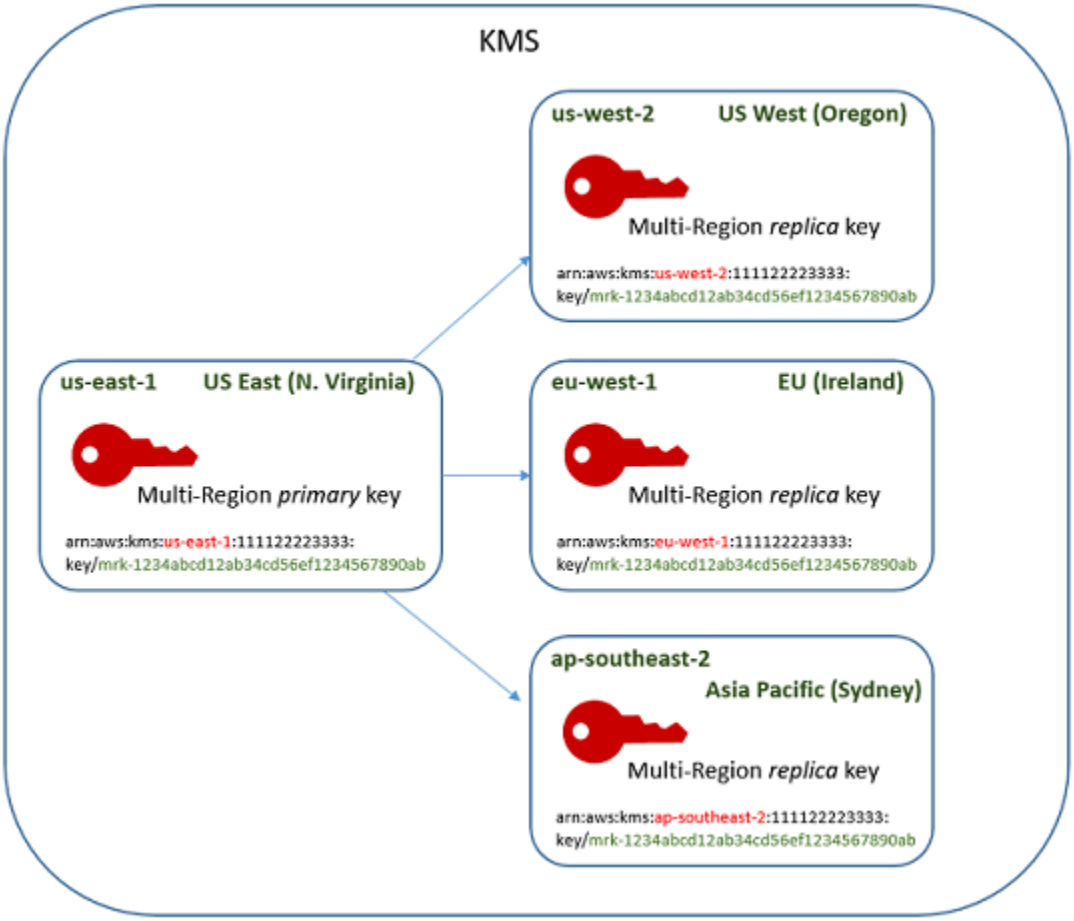
Der resultierende multiregionale Replikatschlüssel ist ein voll funktionsfähiger KMS-Schlüssel mit denselben gemeinsam genutzten Eigenschaften wie der Primärschlüssel. In jeder anderen Hinsicht ist es ein unabhängiger KMS-Schlüssel mit eigener Beschreibung und Schlüsselrichtlinie und den eigenen Erteilungen, Aliasen und Tags. Das Aktivieren oder Deaktivieren eines multiregionalen Schlüssels hat keine Auswirkungen auf verwandte multiregionale Schlüssel. Sie können Primär- und Replikatschlüssel unabhängig von einander in kryptografischen Operationen verwenden oder deren Verwendung koordinieren. Beispielsweise können Sie Daten mit dem Primärschlüssel in der Region USA Ost (Nord-Virginia) verschlüsseln, die Daten in die Region Europa (Irland) verschieben und die Daten mit dem Replikatschlüssel entschlüsseln.

Verwandte multiregionale Schlüssel haben dieselbe Schlüssel-ID. Ihre Schlüssel-ARNs (Amazon Resource Names) unterscheiden sich nur im Feld Region. Zum Beispiel: Der multiregionale Primärschlüssel und die Replikatschlüssel haben die folgenden Beispiel-Schlüssel-ARNs. Die Schlüssel-ID – das letzte Element im Schlüssel ARN – ist identisch. Beide Schlüssel haben die unverwechselbare Schlüssel-ID von multiregionalen Schlüsseln, die mit mrk- beginnt.

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

Für die Interoperabilität ist dieselbe Schlüssel-ID erforderlich. AWS KMS Bindet beim Verschlüsseln die Schlüssel-ID des KMS-Schlüssels an den Chiffretext, sodass der Chiffretext nur mit diesem KMS-Schlüssel oder einem KMS-Schlüssel mit derselben Schlüssel-ID entschlüsselt werden kann. Diese Funktion macht auch verwandte multiregionale Schlüssel leicht zu erkennen und erleichtert die austauschbare Verwendung. Wenn Sie sie beispielsweise in einer Anwendung verwenden, können Sie auf verwandte multiregionale Schlüssel anhand ihrer geteilten Schlüssel-ID verweisen. Geben Sie dann, falls erforderlich, die Region oder den ARN an, um sie zu unterscheiden.

Wenn sich Ihre Datenanforderungen ändern, können Sie den Primärschlüssel auf andere AWS-Regionen in derselben Partition replizieren, z. B. USA West (Oregon) und Asien-Pazifik (Sydney). Das Ergebnis sind vier verwandte multiregionale Schlüssel mit demselben Schlüsselmaterial und denselben Schlüssel-IDs, wie im folgenden Diagramm gezeigt. Sie verwalten die Schlüssel unabhängig voneinander. Sie können sie unabhängig voneinander oder koordiniert verwenden. Beispielsweise können Sie Daten mit dem Replikatschlüssel in Asien-Pazifik (Sydney) verschlüsseln, die Daten nach USA West (Oregon) verschieben und mit dem Replikatschlüssel in USA West (Oregon) entschlüsseln.



Weitere Überlegungen für multiregionalen Schlüssel sind die folgenden:

Synchronisieren gemeinsam genutzter Eigenschaften — Wenn sich eine gemeinsame Eigenschaft der Schlüssel für mehrere Regionen ändert, AWS KMS wird die Änderung vom Primärschlüssel automatisch mit allen zugehörigen Replikatschlüsseln synchronisiert. Sie können eine Synchronisation von gemeinsam genutzten Eigenschaften nicht anfordern oder erzwingen. AWS KMS erkennt und synchronisiert alle Änderungen für Sie. Sie können die Synchronisation

jedoch überprüfen, indem Sie das SynchronizeMultiRegionKeyEreignis in den CloudTrail Protokollen verwenden.

Wenn Sie beispielsweise die automatische Schlüsselrotation für einen symmetrischen Primärschlüssel mit mehreren Regionen aktivieren, wird diese Einstellung auf alle zugehörigen Replikatschlüssel AWS KMS kopiert. Wenn das Schlüsselmaterial gedreht wird, wird die Drehung zwischen allen verwandten multiregionalen Schlüsseln synchronisiert, sodass sie weiterhin dasselbe aktuelle Schlüsselmaterial haben und Zugriff auf alle älteren Versionen des Schlüsselmaterials haben. Wenn Sie einen neuen Replikatschlüssel erstellen, verfügt er über dasselbe aktuelle Schlüsselmaterial aller verwandten multiregionalen Schlüssel und Zugriff auf alle vorherigen Versionen des Schlüsselmaterials. Details hierzu finden Sie unter Drehen von multiregionalen Schlüsseln.

Ändern des Primärschlüssels – Jeder Satz von multiregionalen Schlüsseln muss genau einen Primärschlüssel haben. Der Primärschlüssel ist der einzige Schlüssel, der repliziert werden kann. Er ist auch die Quelle der gemeinsamen Eigenschaften seiner Replikatschlüssel. Sie können jedoch den Primärschlüssel in ein Replikat ändern und einen der Replikatschlüssel als Primärschlüssel heraufstufen. Sie können dies tun, damit Sie einen multiregionalen Primärschlüssel aus einer bestimmten Region löschen oder den Primärschlüssel in einer Region näher an den Projektadministratoren platzieren können. Details hierzu finden Sie unter Aktualisieren der primären Region.

Löschen von Schlüsseln mit mehreren Regionen — Wie bei allen KMS-Schlüsseln müssen Sie das Löschen von Schlüsseln mit mehreren Regionen planen, bevor Sie sie löschen. AWS KMS Solange der Schlüssel zur Löschung ansteht, können Sie ihn nicht in kryptografischen Operationen verwenden. Ein Primärschlüssel mit mehreren Regionen AWS KMS wird jedoch erst gelöscht, wenn alle zugehörigen Replikatschlüssel gelöscht wurden. Details hierzu finden Sie unter Löschen von multiregionalen Schlüsseln.

Konzepte

Die folgenden Begriffe und Konzepte werden mit multiregionalen Schlüsseln verwendet.

Multiregionaler Schlüssel

Ein multiregionaler Schlüssel ist einer von einer Reihe von KMS-Schlüsseln mit derselben Schlüssel-ID und demselben Schlüsselmaterial (und anderen gemeinsam genutzten Eigenschaften) in verschiedenen AWS-Regionen. Jeder multiregionale Schlüssel ist ein voll funktionsfähiger KMS-Schlüssel, der vollständig unabhängig von seinen verwandten multiregionalen Schlüsseln verwendet

werden kann. Da alle zugehörigen Schlüssel für mehrere Regionen dieselbe Schlüssel-ID und dasselbe Schlüsselmaterial haben, sind sie interoperabel, d. h. jeder zugehörige Schlüssel für mehrere Regionen AWS-Region kann Chiffretext entschlüsseln, der mit einem anderen zugehörigen Schlüssel für mehrere Regionen verschlüsselt wurde.

Sie legen die multiregionale Eigenschaft eines KMS-Schlüssels fest, wenn Sie ihn erstellen. Sie können die multiregionale Eigenschaft für einen vorhandenen Schlüssel nicht ändern. Einzelregionale Schlüssel können nicht in multiregionale Schlüssel konvertiert werden oder umgekehrt. Um vorhandene Workloads in multiregionale Szenarien zu verschieben, müssen Sie die Daten erneut verschlüsseln oder neue Signaturen mit neuen multiregionalen Schlüsseln erstellen.

Ein Schlüssel mit mehreren Regionen kann symmetrisch oder asymmetrisch sein und er kann Schlüsselmaterial oder importiertes Schlüsselmaterial verwenden. AWS KMS Sie können keine multiregionale Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden.

In einer Reihe von verwandten multiregionalen Schlüsseln gibt es zu jeder Zeit genau einen Primärschlüssel. Sie können Replikatschlüsseln dieses Primärschlüssels in anderen AWS-Regionen erstellen. Sie können auch die primäre Region aktualisieren, wodurch der Primärschlüssel in einen Replikatschlüssel und ein angegebener Replikatschlüssel in den Primärschlüssel geändert wird. Sie können jedoch jeweils nur einen Primärschlüssel oder Replikatschlüssel verwalten. AWS-Region Alle Regionen müssen sich in derselben AWS -Partition befinden.

Sie können mehrere Sätze von verwandten multiregionalen Schlüsseln in der gleichen oder in unterschiedlichen AWS-Regionen haben. Obwohl verwandte multiregionale Schlüssel interoperabel sind, sind unverwandte multiregionale Schlüssel nicht interoperabel.

Primärschlüssel

Ein Primärschlüssel mit mehreren Regionen ist ein KMS-Schlüssel, der in andere AWS-Regionen in derselben Partition repliziert werden kann. Jeder Satz von multiregionalen Schlüsseln hat nur einen Primärschlüssel.

Ein Primärschlüssel unterscheidet sich von einem Replikatschlüssel wie folgt:

- Nur ein Primärschlüssel kann repliziert werden.
- Der Primärschlüssel ist die Quelle für die gemeinsam genutzten Eigenschaften seiner Replikatschlüssel, einschließlich Schlüsselmaterial und Schlüssel-ID.
- Sie können die automatische Schlüsseldrehung nur für einen Primärschlüssel aktivieren oder deaktivieren.

- Sie können jederzeit das Löschen eines Primärschlüssels planen. Ein Primärschlüssel AWS KMS wird jedoch erst gelöscht, wenn alle seine Replikatschlüssel gelöscht sind.

Primär- und Replikatschlüssel unterscheiden sich jedoch nicht in kryptografischen Eigenschaften. Sie können einen Primärschlüssel und seine Replikatschlüssel austauschbar verwenden.

Sie müssen keinen Primärschlüssel replizieren. Sie können ihn genauso verwenden wie jeden KMS-Schlüssel und dabb replizieren, wenn es nützlich ist. Da multiregionale Schlüssel jedoch andere Sicherheitseigenschaften haben als einzelregionale Schlüssel, empfiehlt es sich, einen multiregionalen Schlüssel nur dann zu erstellen, wenn Sie ihn replizieren möchten.

Replikat-Schlüssel

Ein multiregionaler Replikatschlüssel ist ein KMS-Schlüssel, der die gleiche Schlüssel-ID und das gleiche Schlüsselmaterial wie sein Primärschlüssel und seine verwandten Replikatschlüsseln hat, aber in einer anderen AWS-Region existiert.

Ein Replikatschlüssel ist ein voll funktionsfähiger KMS-Schlüssel mit eigener Schlüsselrichtlinie und den eigenen Erteilungen, Aliasen, Tags und anderen Eigenschaften. Es handelt sich nicht um eine Kopie oder einen Zeiger auf den Primärschlüssel oder einen sonstigen Schlüssel. Sie können einen Replikatschlüssel auch dann verwenden, wenn sein Primärschlüssel und alle verwandte Replikatschlüssel deaktiviert sind. Sie können auch einen Replikatschlüssel in einen Primärschlüssel und einen Primärschlüssel in einen Replikatschlüssel konvertieren. Sobald er erstellt wurde, stützt sich ein Replikatschlüssel auf seinen Primärschlüssel nur zur Schlüsseldrehung und zum Aktualisieren der primären Region.

Primär- und Replikatschlüssel unterscheiden sich nicht in kryptografischen Eigenschaften. Sie können einen Primärschlüssel und seine Replikatschlüssel austauschbar verwenden. Daten, die durch einen Primär- oder Replikatschlüssel verschlüsselt werden, können durch denselben Schlüssel oder durch einen beliebigen verwandten Primär- oder Replikatschlüssel entschlüsselt werden.

Replicate

Sie können einen Primärschlüssel mit mehreren Regionen in einen anderen AWS-Region Primärschlüssel in derselben Partition replizieren. Wenn Sie dies tun, AWS KMS wird in der angegebenen Region ein Replikatschlüssel mit mehreren Regionen erstellt, der dieselbe Schlüssel-ID und andere gemeinsame Eigenschaften wie der Primärschlüssel hat. Dann transportiert es das Schlüsselmaterial sicher über die Regionsgrenze und ordnet es dem neuen Replikatschlüssel zu, alles innerhalb von AWS KMS.

Freigegebene Eigenschaften

Gemeinsam genutzte Eigenschaften sind Eigenschaften eines Primärschlüssels mit mehreren Regionen, die mit seinen Replikatschlüsseln gemeinsam genutzt werden. AWS KMS erstellt die Replikatschlüssel mit denselben gemeinsamen Eigenschaftswerten wie die des Primärschlüssels. Anschließend synchronisiert es regelmäßig die gemeinsamen Eigenschaftswerte des Primärschlüssels mit seinen Replikatschlüsseln. Sie können diese Eigenschaften nicht für einen Replikatschlüssel festlegen.

Im Folgenden sind die gemeinsamen Eigenschaften von multiregionalen Schlüsseln aufgeführt.

- Schlüssel-ID – (Das `Region`-Element des Schüssel-ARN unterscheidet sich.)
- Schlüsselmaterial
- Ursprung des Schlüsselmaterials
- Schlüsselspezifikation und Verschlüsselungsalgorithmen
- Schlüsselnutzung
- Automatische Schlüsseldrehung – Sie können die automatische Schlüsseldrehung nur für den Primärschlüssel aktivieren oder deaktivieren. Neue Replikatschlüssel werden mit allen Versionen des freigegebenen Schlüsselmaterials erstellt. Details hierzu finden Sie unter Drehen von multiregionalen Schlüsseln.
- Rotation auf Anforderung — Sie können die Rotation auf Anforderung nur für den Primärschlüssel durchführen. Neue Replikatschlüssel werden mit allen Versionen des freigegebenen Schlüsselmaterials erstellt. Details hierzu finden Sie unter Drehen von multiregionalen Schlüsseln.

Sie können sich auch die Primär- und Replikatbezeichnungen verwandter Mehrbereichsschlüssel als gemeinsame Eigenschaften vorstellen. Wenn Sie neue Replikatschlüssel erstellen oder den Primärschlüssel aktualisieren, wird die Änderung mit allen zugehörigen Multiregions-Schlüsseln AWS KMS synchronisiert. Wenn diese Änderungen abgeschlossen sind, werden alle verwandte multiregionale Schlüssel ihren Primärschlüssel und die Replikatschlüssel korrekt auflisten.

Alle anderen Eigenschaften von multiregionalen Schlüsseln sind unabhängige Eigenschaften, einschließlich der Beschreibung, Schlüsselrichtlinie, Erteilungen, aktivierten und deaktivierten Schlüsselzustände, Aliasen und Tags. Sie können dieselben Werte für diese Eigenschaften für alle verwandte mutliregionale Schlüssel festlegen. Wenn Sie jedoch den Wert einer unabhängigen Eigenschaft ändern, wird dies von AWS KMS nicht synchronisiert.

Sie können die Synchronisierung der freigegebenen Eigenschaften Ihrer multiregionalen Schlüssel verfolgen. Suchen Sie in Ihrem AWS CloudTrail Protokoll nach dem Ereignis. SynchronizeMultiRegionKey

Steuern des Zugriffs auf multiregionale Schlüssel

Sie können multiregionale Schlüssel in Compliance-, Notfallwiederherstellungs- und Backup-Szenarien verwenden, die mit einzelregionalen Schlüsseln komplexer wären. Da sich die Sicherheitseigenschaften von multiregionalen Schlüsseln erheblich von denen von einzelregionalen Schlüsseln unterscheiden, empfehlen wir, bei der Autorisierung der Erstellung, Verwaltung und Verwendung von multiregionalen Schlüsseln Vorsicht walten zu lassen.

> **Note**
>
> Vorhandene IAM-Richtlinienanweisungen mit Platzhalterzeichen im `Resource`-Feld gelten jetzt sowohl für einzelregionale Schlüssel als auch für multiregionale Schlüssel. Um sie auf einzelregionale KMS-Schlüssel oder multiregionale Schlüssel zu beschränken, verwenden Sie den Bedingungsschlüssel kms:MultiRegion.

Verwenden Sie Ihre Autorisierungstools, um die Erstellung und Verwendung von multiregionalen Schlüsseln in jedem Szenario zu verhindern, in dem ein einzelregionaler Schlüssel ausreicht. Erlauben Sie Prinzipalen die Replikation eines multiregionalen Schlüssels nur in AWS-Regionen, die sie erfordern. Erteilen Sie Berechtigungen für multiregionale Schlüssel nur an Prinzipale, die sie benötigen, und nur für Aufgaben, für die sie erforderlich sind.

Sie können Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen verwenden, um IAM-Prinzipalen die Verwaltung und Verwendung von multiregionalen Schlüsseln in Ihrem AWS-Konto zu erlauben. Jeder multiregionaler Schlüssel ist eine unabhängige Ressource mit einem eindeutigen Schlüssel-ARN und einer Schlüsselrichtlinie. Sie müssen für jeden Schlüssel eine Schlüsselrichtlinie einrichten und pflegen und sicherstellen, dass neue und vorhandene IAM-Richtlinien Ihre Autorisierungsstrategie implementieren.

Themen

- Grundlagen der Autorisierung für multiregionale Schlüssel
- Autorisieren von Administratoren und Benutzern für multiregionale Schlüssel
- Autorisieren von AWS KMS zum Synchronisieren der multiregionalen Schlüsseln

Grundlagen der Autorisierung für multiregionale Schlüssel

Berücksichtigen Sie beim Entwerfen von Schlüsselrichtlinien und IAM-Richtlinien für multiregionale Schlüssel die folgenden Grundsätze.

- Schlüsselrichtlinie – Jeder multiregionale Schlüssel ist eine unabhängige KMS-Schlüsselressource mit einer eigenen Schlüsselrichtlinie. Sie können dieselbe oder eine andere Schlüsselrichtlinie auf jeden Schlüssel in dem Satz von verwandten multiregionalen Schlüsseln anwenden. Schlüsselrichtlinien sind nicht gemeinsam genutzte Eigenschaftenvon multiregionalen Schlüsseln. AWS KMS kopiert oder synchronisiert Schlüsselrichtlinien zwischen verwandten multiregionalen Schlüsseln nicht.

  Wenn Sie einen Replikatschlüssel in der AWS KMS-Konsole erstellen, zeigt die Konsole die aktuelle Schlüsselrichtlinie des Primärschlüssels als Hilfe an. Sie können diese Schlüsselrichtlinie verwenden, bearbeiten oder löschen und ersetzen. Aber selbst wenn Sie die Primärschlüsselrichtlinie unverändert akzeptieren, synchronisiert AWS KMS die Richtlinien nicht. Wenn Sie beispielsweise die Schlüsselrichtlinie des Primärschlüssels ändern, bleibt die Schlüsselrichtlinie des Replikatschlüssels unverändert.
- Standardschlüsselrichtlinie – Wenn Sie multiregionale Schlüssel mithilfe der `ReplicateKey` Operationen CreateKey und erstellen, wird die Standardschlüsselrichtlinie angewendet, es sei denn, Sie geben eine Schlüsselrichtlinie in der Anforderung an. Dies ist die gleiche Standard-Schlüsselrichtlinie, die auf einzelregionale Schlüssel angewendet wird.
- IAM-Richtlinien – Wie bei allen KMS-Schlüsseln können Sie IAM-Richtlinien verwenden, um den Zugriff auf multiregionale Schlüssel nur dann zu steuern, wenn die Schlüsselrichtlinie es erlaubt. IAM-Richtlinien – gelten standardmäßig für alle AWS-Regionen. Sie können jedoch Bedingungsschlüssel wie aws:RequestedRegion verwenden, um Berechtigungen auf eine bestimmte Region zu beschränken.

  Zum Erstellen von Primär- und Replikatschlüsseln müssen Prinzipale die `kms:CreateKey`-Berechtigung in einer IAM-Richtlinie haben, die für die Region gilt, wo der Schlüssel erstellt wird.
- Erteilungen – AWS KMS-Erteilungen sind regional. Jede Erteilung erlaubt Berechtigungen für einen KMS-Schlüssel. Sie können Erteilungen verwenden, um Berechtigungen für einen multiregionalen Primärschlüssel oder Replikatschlüssel zuzulassen. Sie können jedoch keine einzelne Erteilung verwenden, um Berechtigungen für mehrere KMS-Schlüssel zuzulassen, selbst wenn es sich um verwandte multiregionale Schlüssel handelt.

- Schüssel-ARN – Jeder multiregionale Schlüssel verfügt über einen Schüssel-ARN. Die Schlüssel-ARNs von verwandten multiregionalen Schlüsseln haben die gleiche Partition und Schlüssel-ID und das gleiche Konto, aber unterschiedliche Regionen.

  Um eine IAM-Richtlinienanweisung auf einen bestimmten multiregionalen Schlüssel anzuwenden, verwenden Sie dessen Schlüssel-ARN oder ein Schlüssel-ARN-Muster, das die Region enthält. Verwenden Sie ein Platzhalterzeichen (*) im Region-Element des ARN, um eine IAM-Richtlinienanweisung auf alle verwandten multiregionalen Schlüssel.

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

  Um eine Richtlinienanweisung auf alle multiregionalen Schlüssel in Ihrem anzuwendenAWS-Konto, können Sie die kms:MultiRegion-Richtlinienbedingung oder ein Schlüssel-ID-Muster verwenden, das das einzigartige `mrk-` Präfix enthält.

- Serviceverknüpfte Rolle – Prinzipale, die multiregionale Primärschlüssel erstellen, müssen über die Berechtigung iam:CreateServiceLinkedRole verfügen.

  So synchronisieren Sie die gemeinsam genutzten Eigenschaften verwandter multiregionaler Schlüssel: AWS KMS übernimmt eine serviceverknüpfte IAM-Rolle. AWS KMS erstellt die serviceverknüpfte Rolle im AWS-Konto, jedes Mal wenn Sie einen multiregionalen Primärschlüssel erstellen. (Wenn die Rolle vorhanden ist, erstellt AWS KMS sie neu, was keine schädliche Wirkung hat.) Die Rolle ist in allen Regionen gültig. Damit die serviceverknüpfte Rolle AWS KMS erstellen (oder neu erstellen) kann, müssen Prinzipale, die multiregionale Primärschlüssel erstellen, über die Berechtigung iam:CreateServiceLinkedRole verfügen.

Autorisieren von Administratoren und Benutzern für multiregionale Schlüssel

Prinzipale, die multiregionale Schlüssel erstellen und verwalten, benötigen die folgenden Berechtigungen in den primären und Replikatregionen:

- `kms:CreateKey`
- `kms:ReplicateKey`
- `kms:UpdatePrimaryRegion`
- `iam:CreateServiceLinkedRole`

Erstellen eines Primärschlüssels

Um einen multiregionalen Primärschlüssel zu erstellen, benötigt der Prinzipal kms:CreateKey- und iam:CreateServiceLinkedRole-Berechtigungen in einer IAM-Richtlinie, die in der Region des Primärschlüssels wirksam ist. Prinzipale, die über diese Berechtigungen verfügen, können einzelregionale Schlüssel und multiregionale Schlüssel erstellen, es sei denn, Sie beschränken ihre Berechtigungen.

Die -`iam:CreateServiceLinkedRole`Berechtigung ermöglicht es AWS KMS, die AWSServiceRoleForKeyManagementServiceMultiRegionKeys Rolle zum Synchronisieren der gemeinsamen Eigenschaften verwandter multiregionaler Schlüssel zu erstellen.

Zum Beispiel kann mit dieser IAM-Richtlinie ein Prinzipal einen beliebigen KMS-Schlüsseltyp erstellen.

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

Um die Berechtigung zum Erstellen von multiregionalen Primärschlüsseln zu gewähren oder zu verweigern, verwenden Sie den kms:MultiRegion-Bedingungsschlüssel. Gültige Werte sind `true` (multiregionaler Schlüssel) oder `false` (einzelregionaler Schlüssel). Beispielsweise verwendet die folgende IAM-Richtlinienanweisung eine `Deny`-Aktion mit dem `kms:MultiRegion`-Bedingungsschlüssel, um zu verhindern, dass Prinzipale multiregionale Schlüssel erstellen.

```
{
```

```
  "Version": "2012-10-17",
  "Statement":{
      "Action":"kms:CreateKey",
      "Effect":"Deny",
      "Resource":"*",
      "Condition": {
          "Bool": "kms:MultiRegion": true
      }
  }
}
```

Replizieren von Schlüsseln

Um einen multiregionalen Replikatschlüssel zu erstellen, benötigt der Prinzipal die folgenden Berechtigungen:

- kms:ReplicateKey Berechtigung in der Schlüsselrichtlinie des Primärschlüssels.
- kms:CreateKey Berechtigung in einer IAM-Richtlinie, die in der Region des Replikatschlüssels wirksam ist.

Seien Sie vorsichtig, wenn Sie diese Berechtigungen zulassen. Sie ermöglichen es Prinzipalen, KMS-Schlüssel und die Schlüsselrichtlinien zu erstellen, die ihre Verwendung autorisieren. Die `kms:ReplicateKey`-Berechtigung autorisiert auch die Übertragung von Schlüsselmaterial über Regionsgrenzen innerhalb von AWS KMS.

Um die einzuschränken, AWS-Regionen in der ein multiregionaler Schlüssel repliziert werden kann, verwenden Sie den Bedingungsschlüssel kms:ReplicaRegion. Es begrenzt nur die`kms:ReplicateKey`-Berechtigung. Andernfalls hat es keine Auswirkungen. Beispielsweise ermöglicht die folgende Schlüsselrichtlinie dem Prinzipal, diesen Primärschlüssel zu replizieren, jedoch nur in den angegebenen Regionen.

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

Aktualisieren der primären Region

Autorisierte Prinzipale können einen Replikatschlüssel in einen Primärschlüssel konvertieren, wodurch der frühere Primärschlüssel in ein Replikat umgewandelt wird. Diese Aktion ist bekannt als Aktualisieren der primären Region. Um die primäre Region zu aktualisieren, benötigt der Prinzipal die kms:-UpdatePrimaryRegionBerechtigung in beiden Regionen. Sie können diese Berechtigungen in einer Schlüsselrichtlinie oder einer IAM-Richtlinie bereitstellen.

- `kms:UpdatePrimaryRegion` für den Primärschlüssel. Diese Berechtigung muss in der Region des Primärschlüssels wirksam sein.
- `kms:UpdatePrimaryRegion` für den Replikatschlüssel. Diese Berechtigung muss in der Region des Replikatschlüssels wirksam sein.

Die folgende Schlüsselrichtlinie gibt Benutzern, die die Administratorrolle übernehmen können, die Berechtigung zum Aktualisieren der primären Region des KMS-Schlüssels. Dieser KMS-Schlüssel kann der Primärschlüssel oder ein Replikatschlüssel in dieser Operation sein.

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

Um die einzuschränkenAWS-Regionen, die einen Primärschlüssel hosten kann, verwenden Sie den kms:PrimaryRegion-Bedingungsschlüssel. Mit der folgenden IAM-Richtlinienanweisung können die Prinzipale beispielsweise die primäre Region der multiregionalen Schlüssel im AWS-Konto aktualisieren, jedoch nur, wenn die neue primäre Region eine der angegebenen Regionen ist.

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
      "arn:aws:kms:*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-west-2",
         "sa-east-1",
         "ap-southeast-1"
      ]
    }
  }
}
```

Verwenden und Verwalten von multiregionalen Schlüsseln

Standardmäßig haben Prinzipale, die über die Berechtigung zum Verwenden und Verwalten von KMS-Schlüsseln in einem AWS-Konto und einer Region haben, auch die Berechtigung, multiregionale Schlüssel zu verwenden und zu verwalten. Sie können jedoch den Bedingungsschlüssel kms:MultiRegion verwenden, um nur einzelregionale Schlüssel oder nur multiregionale Schlüssel zuzulassen. Oder verwenden Sie den kms:MultiRegionKeyType-Bedingungsschlüssel, um nur multiregionale Primärschlüssel oder nur Replikatschlüssel zuzulassen. Beide Bedingungsschlüssel steuern den Zugriff auf die -CreateKeyOperation und auf jede Operation, die einen vorhandenen KMS-Schlüssel verwendet, z. B. Encrypt oder EnableKey.

Beispielsweise verwendet die folgende IAM-Richtlinienanweisung den `kms:MultiRegion`-Bedingungsschlüssel, um zu verhindern, dass Prinzipale multiregionale Schlüssel verwenden oder verwalten.

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

In diesem Beispiel verwendet die IAM-Richtlinienanweisung die `kms:MultiRegionKeyType`-Bedingung, damit Prinzipale das Löschen von Schlüsseln planen und abbrechen können, jedoch nur für multiregionale Replikatschlüssel.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
      "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Autorisieren von AWS KMS zum Synchronisieren der multiregionalen Schlüsseln

Zur Unterstützung von multiregionalen Schlüsseln, verwendet AWS KMS eine serviceverknüpfte IAM-Rolle. Diese Rolle gibt AWS KMS die Berechtigungen, die es zum Synchronisieren von gemeinsam genutzten Eigenschaften benötigt. Sie können das SynchronizeMultiRegionKey CloudTrail Ereignis anzeigen, das Datensätze aufzeichnet, die freigegebene Eigenschaften in Ihren AWS CloudTrail Protokollen AWS KMS synchronisieren.

Über die serviceverknüpfte Rolle für multiregionale Schlüssel

Eine serviceverknüpfte Rolle ist eine IAM-Rolle, die einem AWS-Service die Berechtigung gewährt, in Ihrem Namen andere AWS-Services aufzurufen. Sie soll die Verwendung der Funktionen mehrerer integrierter AWS-Services für Sie vereinfachen, da Sie keine komplexen IAM-Richtlinien erstellen und verwalten müssen.

Bei multiregionalen Schlüsseln AWS KMS erstellt die AWSServiceRoleForKeyManagementServiceMultiRegionKeys serviceverknüpfte Rolle mit der AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy Richtlinie . Diese Richtlinie gibt der Rolle die `kms:SynchronizeMultiRegionKey`-Berechtigung, die es ermöglicht, die freigegebenen Eigenschaften von multiregionalen Schlüsseln zu synchronisieren.

Da die AWSServiceRoleForKeyManagementServiceMultiRegionKeys serviceverknüpfte Rolle nur vertraut`mrk.kms.amazonaws.com`, AWS KMS kann nur diese serviceverknüpfte Rolle übernehmen.

Diese Rolle ist auf die Vorgänge beschränkt, die AWS KMS benötigt, um freigegebene multiregionale Eigenschaften zu synchronisieren. AWS KMS werden keine weiteren Berechtigungen erteilt. Zum Beispiel hat AWS KMS nicht die Berechtigung zum Erstellen, Replizieren oder Löschen von KMS-Schlüsseln.

Weitere Informationen darüber, wie AWS-Services serviceverknüpften Rollen verwenden, finden Sie unter Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch.

Erstellen der serviceverknüpften Rolle

AWS KMS erstellt automatisch die AWSServiceRoleForKeyManagementServiceMultiRegionKeys serviceverknüpfte Rolle in Ihrem , AWS-Konto wenn Sie einen multiregionalen Schlüssel erstellen, sofern die Rolle noch nicht vorhanden ist. Sie können diese serviceverknüpfte Rolle nicht direkt erstellen oder direkt neu erstellen.

Bearbeiten der Beschreibung der serviceverknüpften Rolle

Sie können den Rollennamen oder die Richtlinienanweisungen in der AWSServiceRoleForKeyManagementServiceMultiRegionKeys serviceverknüpften Rolle nicht bearbeiten, aber Sie können die Rollenbeschreibung bearbeiten. Anweisungen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen der serviceverknüpften Rolle

AWS KMS löscht die AWSServiceRoleForKeyManagementServiceMultiRegionKeys serviceverknüpfte Rolle nicht aus Ihrem AWS-Konto und Sie können sie nicht löschen. übernimmt jedoch AWS KMS die AWSServiceRoleForKeyManagementServiceMultiRegionKeys Rolle und verwendet keine ihrer Berechtigungen, es sei denn, Sie haben multiregionale Schlüssel in Ihrem AWS-Konto und Ihrer Region.

Erstellen von multiregionalen Schlüsseln

Sie können in der Konsole multiregionale Schlüssel erstellen, oder mithilfe der AWS KMS-API.

Die multiregionale Eigenschaft, die Sie in diesem Verfahren festgelegt haben, ist unveränderlich. Einzelregionale Schlüssel können nicht in multiregionale Schlüssel konvertiert werden oder umgekehrt.

Themen

- Multiregionale Primärschlüssel erstellen
- Erstellen von multiregionalen Replikatschlüsseln

Multiregionale Primärschlüssel erstellen

Sie können einen multiregionalen Primärschlüssel in der AWS KMS-Konsole erstellen, oder mithilfe der AWS KMS-API. Sie können den Primärschlüssel in jeder AWS-Region erstellen, in der AWS KMS multiregionale Schlüssel unterstützt.

Um einen multiregionalen Primärschlüssel zu erstellen, benötigt der Prinzipal die gleichen Berechtigungen, die er zum Erstellen eines beliebigen KMS-Schlüssels benötigt, einschließlich der kms:CreateKey-Berechtigung in einer IAM-Richtlinie. Der Prinzipal benötigt auch die iam:CreateServiceLinkedRole-Berechtigung. Sie können den Bedingungsschlüssel kms:MultiRegionKeyType verwenden, um die Berechtigung zum Erstellen von multiregionalen Primärschlüsseln zu gewähren oder zu verweigern.

Mit diesen Anweisungen wird ein multiregionaler Primärschlüssel mit Schlüsselmaterial erstellt, das AWS KMS generiert. Weitere Informationen zum Erstellen eines multiregionalen Primärschlüssels mit importiertem Schlüsselmaterial finden Sie unter Erstellen eines Primärschlüssels mit importiertem Schlüsselmaterial.

Themen

- Einen multiregionalen Primärschlüssel erstellen (Konsole)
- Einen multiregionalen Primärschlüssel erstellen (AWS KMS-API)

Einen multiregionalen Primärschlüssel erstellen (Konsole)

Zum Erstellen eines multiregionalen Primärschlüssels in der AWS KMS-Konsole verwenden Sie den gleichen Prozess, den Sie zum Erstellen eines beliebigen KMS-Schlüssels verwenden würden. Wählen Sie einen multiregionalen Schlüssel unter Erweiterte Optionen aus. Vollständige Anweisungen finden Sie unter Erstellen von Schlüsseln.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Wählen Sie einen symmetrischen oder asymmetrischen Schlüsseltyp. Symmetrische Schlüssel sind die Standardeinstellung.

   Sie können multiregionale symmetrische und asymmetrische Schlüssel, einschließlich multiregionaler HMAC-KMS-Schlüssel, die symmetrisch sind, erstellen.
6. Wählen Sie Ihre Schlüsselverwendung aus. Encrypt and decrypt (Verschlüsseln und Entschlüsseln) ist die Standardeinstellung.

   Weitere Informationen finden Sie unter the section called “Erstellen von Schlüsseln”, the section called “Erstellen asymmetrischer KMS-Schlüssel” oder the section called “Erstellen von HMAC-Schlüsseln”.
7. Erweitern Sie Advanced options (Erweiterte Optionen).
8. Unter Ursprung des Schlüsselmaterials, damit AWS KMS das Schlüsselmaterial generiert, das Ihre Primär- und Replikatschlüssel gemeinsam nutzen, wählen Sie KMS aus. Wenn Sie Schlüsselmaterial in die Primär- oder die Replikatschlüssel importieren, wählen Sie External (Extern) aus.
9. Unter Multi-Region replication (multiregionale Replikation), wählen Sie Allow this key to be replicated into other Regions (Replizieren dieses Schlüssels in andere Regionen zulassen) aus.

   Sie können diese Einstellung nicht ändern, nachdem Sie den KMS-Schlüssel erstellt haben.
10. Geben Sie einen Alias für den Primärschlüssel ein.

    Aliasse sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten denselben Alias oder verschiedene Aliasse zuweisen. AWS KMS synchronisiert die Aliasse von multiregionalen Schlüsseln nicht.

    > **Note**
    >
    > Wenn Sie einen Alias hinzufügen, löschen oder aktualisieren, wird dadurch möglicherweise eine Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details

dazu finden Sie unter ABAC für AWS KMS und Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel.

11. (Optional) Geben Sie eine Beschreibung für den Primärschlüssel ein.

    Beschreibungen sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten dieselbe Beschreibung oder verschiedene Beschreibungen geben. AWS KMS synchronisiert die Beschreibungen von multiregionalen Schlüsseln nicht.

12. (Optional) Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wenn Sie dem Primärschlüssel mehrere Tags zuweisen möchten, wählen Sie Add tag (Tag hinzufügen) aus.

    Tags sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten dieselben Tags oder verschiedene Tags zuweisen. AWS KMS synchronisiert die Tags von multiregionalen Schlüsseln nicht. Sie können die Tags auf KMS-Schlüssel jederzeit ändern.

    > **Note**
    >
    > Wenn Sie einen KMS-Schlüssel markieren oder entmarkieren, wird dadurch möglicherweise die Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.

13. Wählen Sie die IAM-Benutzer und -Rollen aus, die den KMS-Schlüssel verwalten können.

    > **Note**
    >
    > IAM-Richtlinien können anderen IAM-Benutzern und -Rollen die erforderlichen Berechtigungen zum Verwalten des KMS-Schlüssels erteilen.
    > Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

    Dieser Schritt startet den Prozess zum Erstellen einer Schlüsselrichtlinie für den Primärschlüssel. Schlüsselrichtlinien sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln.

Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten dieselbe Schlüsselrichtlinie oder verschiedene Schlüsselrichtlinien zuweisen. AWS KMS synchronisiert die Schlüsselrichtlinien von multiregionalen Schlüsseln nicht. Sie können die Schlüsselrichtlinie eines KMS-Schlüssels jederzeit ändern.

14. Führen Sie die Schritte zum Erstellen der Schlüsselrichtlinie durch, einschließlich der Auswahl von Schlüsselbenutzern. Wählen Sie nach dem Überprüfen der Schlüsselrichtlinie Finish (beenden) aus, um den KMS-Schlüssel zu erstellen.

Einen multiregionalen Primärschlüssel erstellen (AWS KMS-API)

Um einen multiregionalen Primärschlüssel zu erstellen, verwenden Sie die -CreateKeyOperation. Sie müssen außerdem den `MultiRegion`-Parameter mit dem Wert `True` verwenden.

Beispielsweise erstellt der folgende Befehl einen multiregionalen Primärschlüssel in der AWS-Region des Anrufers (us-east-1). Er akzeptiert Standardwerte für alle anderen Eigenschaften, einschließlich der Schlüsselrichtlinie. Die Standardwerte für multiregionale Primärschlüssel sind dieselben wie die Standardwerte für alle anderen KMS-Schlüssel, einschließlich der Standard-Schlüsselrichtlinie. Dieses Verfahren erstellt einen symmetrischen Verschlüsselungsschlüssel, den Standard-KMS-Schlüssel.

Die Antwort enthält das `MultiRegion`-Element und das `MultiRegionConfiguration`-Element mit typischen Unterelementen und Werten für einen multiregionalen Primärschlüssel ohne Replikatschlüssel. Die Schlüssel-ID eines multiregionalen Schlüssels beginnt immer mit `mrk-`.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

```
$ 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": [ ]
      }
    }
}
```

Erstellen von multiregionalen Replikatschlüsseln

Sie können einen multiregionalen Replikatschlüssel in der -AWS KMSKonsole, mithilfe der AWS CloudFormation-ReplicateKeyOperation oder mithilfe einer Vorlage erstellen. Sie können die -CreateKeyOperation nicht verwenden, um einen Replikatschlüssel zu erstellen.

Sie können diese Verfahren verwenden, um jeden multiregionalen Primärschlüssel zu replizieren, einschließlich ein KMS-Schlüssel mit symmetrischer Verschlüsselung, ein asymmetrischer KMS-Schlüssel oder einen HMAC-KMS-Schlüssel.

Wenn diese Produktion abgeschlossen ist, verfügt der neue Replikatschlüssel über einen vorübergehenden Schlüsselstatus von `Creating`. Dieser Schlüsselstatus ändert sich nach einigen Sekunden in `Enabled` (oder PendingImport), wenn der Prozess der Erstellung des neuen Replikatschlüssels abgeschlossen ist. Während der Schlüsselstatus `Creating` ist, können Sie den Schlüssel verwalten, Sie können ihn jedoch noch nicht in kryptografischen Operationen verwenden. Wenn Sie den Replikatschlüssel programmgesteuert erstellen und verwenden, versuchen Sie es erneut `KMSInvalidStateException` oder rufen Sie auf, um seinen `KeyState` Wert DescribeKey zu überprüfen, bevor Sie ihn verwenden.

Wenn Sie versehentlich einen Replikatschlüssel löschen, können Sie ihn mit diesem Verfahren neu erstellen. Wenn Sie denselben Primärschlüssel in derselben Region replizieren, hat der neu erstellte Replikatschlüssel dieselben gemeinsamen Eigenschaften wie der ursprüngliche Replikatschlüssel.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Weitere Informationen

- Weitere Informationen zum Erstellen eines multiregionalen Replikatschlüssels mit importiertem Schlüsselmaterial finden Sie unter Erstellen eines Replikatschlüssels mit importiertem Schlüsselmaterial.
- Informationen zur Verwendung einer AWS CloudFormation Vorlage zum Erstellen eines Replikatschlüssels finden Sie AWS::KMS::ReplicaKey unter im AWS CloudFormation - Benutzerhandbuch.

Themen

- Replikatregionen
- Erstellen von Replikatschlüsseln (Konsole)
- Erstellen eines Replikatschlüssels (AWS KMS-API)

Replikatregionen

Normalerweise replizieren Sie einen multiregionalen Schlüssel in eine AWS-Region, die Ihrem Geschäftsmodell und den regulatorischen Anforderungen entspricht. Sie können beispielsweise einen Schlüssel in Regionen replizieren, in denen Sie Ihre Ressourcen behalten. Oder Sie könnten, um eine Anforderung zur Notfallwiederherstellung zu erfüllen, einen Schlüssel in geografisch entfernte Regionen replizieren.

Folgendes sind die AWS KMS-Anforderungen für Replikatregionen. Wenn die ausgewählte Region diese Anforderungen nicht erfüllt, schlagen Versuche, einen Schlüssel zu replizieren, fehl.

- Ein verwandter multiregionaler Schlüssel pro Region – Sie können keinen Replikatschlüssel in derselben Region wie sein Primärschlüssel oder in derselben Region wie ein anderes Replikat des Primärschlüssels erstellen.

  Wenn Sie versuchen, einen Primärschlüssel in einer Region zu replizieren, die bereits über ein Replikat dieses Primärschlüssels verfügt, schlägt der Versuch fehl. Wenn sich der aktuelle Replikationsschlüssel in der Region im `PendingDeletion`-Schlüsselstatus befindet, können Sie die Löschung des Replikationsschlüssels abbrechen oder warten, bis der Replikationsschlüssel gelöscht ist.
- Mehrere nicht verwandte multiregionale Schlüssel in derselben Region – Sie können mehrere nicht verwandte multiregionale Schlüssel in derselben Region haben. Beispielsweise können Sie zwei multiregionale Primärschlüssel in der Region `us-east-1` haben. Jeder Primärschlüssel kann einen Replikatschlüssel in der Region `us-west-2` haben.
- Regionen in derselben Partition – Die Region des Replikatschlüssels muss sich in derselben AWS-Partition befinden, wie die Region des Primärschlüssels.
- Region muss aktiviert sein – Wenn eine Region standardmäßig deaktiviert ist, können Sie in dieser Region keine Ressourcen erstellen, bis sie für Ihr AWS-Konto aktiviert ist.

Erstellen von Replikatschlüsseln (Konsole)

In der AWS KMS-Konsole können Sie ein oder mehrere Replikate eines multiregionalen Primärschlüssels in der gleichen Produktion erstellen.

Dieses Verfahren ähnelt dem Erstellen eines einzelregionalen KMS-Schlüssels in der Konsole. Da jedoch ein Replikatschlüssel auf dem Primärschlüssel basiert, wählen Sie keine Werte für gemeinsam genutzte Eigenschaften aus, z. B. die Schlüsselspezifikation (symmetrisch oder asymmetrisch), die Schlüsselnutzung oder den Schlüsselursprung.

Sie geben jedoch Eigenschaften an, die nicht freigegeben werden, einschließlich eines Alias, Tags, einer Beschreibung und einer Schlüsselrichtlinie. Die Konsole zeigt die aktuellen Eigenschaftswerte des Primärschlüssels an, Sie können sie jedoch ändern. Selbst wenn Sie die Primärschlüsselwerte beibehalten, synchronisiert AWS KMS diese Werte nicht.

> **Important**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie die Schlüssel-ID oder den Alias eines multiregionalen Primärschlüssels aus. Dadurch wird die Seite mit den Schlüsseldetails des KMS-Schlüssels geöffnet.

   Um einen multiregionalen Primärschlüssel zu identifizieren, verwenden Sie das Werkzeugsymbol in der oberen rechten Ecke, um die Spalte Regionality (Regionalität) zur Tabelle hinzuzufügen.
5. Wählen Sie die Registerkarte Regionality (Regionalität) aus.
6. Unter Related multi-Region keys (Verwandte multiregionale Schlüssel) wählen Sie Create new replica keys (neue Replikatschlüssel erstellen) aus.

   Der Abschnitt Related multi-Region keys (Verwandte multiregionale Schlüssel) zeigt die Region des Primärschlüssels und seiner Replikatschlüssel an. Sie können diese Anzeige verwenden, um die Region für den neuen Replikatschlüssel auszuwählen.
7. Wählen Sie mindestens eine AWS-Regionen aus. Bei diesem Verfahren wird in jedem der ausgewählten Regionen ein Replikatschlüssel erstellt.

   Das Menü enthält nur Regionen in derselben AWS-Partition als der Primärschlüssel. Regionen, die bereits über einen verwandten multiregionalen Schlüssel verfügen, werden angezeigt, können jedoch nicht ausgewählt werden. Möglicherweise sind Sie nicht berechtigt, einen Schlüssel in alle Regionen im Menü zu replizieren.

   Wenn Sie die Auswahl von Regionen abgeschlossen haben, schließen Sie das Menü. Die ausgewählten Regionen werden angezeigt. Um die Replikation in eine Region abzubrechen, wählen Sie das Kontrollkästchen X neben dem Namen der Region.
8. Geben Sie einen Alias für den Replikatschlüssel ein.

Die Konsole zeigt einen der aktuellen Aliasen des Primärschlüssels an, Sie können ihn jedoch ändern. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten denselben Alias oder verschiedene Aliase zuweisen. Aliase sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. AWS KMS synchronisiert die Aliase von multiregionalen Schlüsseln nicht.

Wenn Sie einen Alias hinzufügen, löschen oder aktualisieren, wird dadurch möglicherweise eine Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Aliassen zur Steuerung des Zugriffs auf KMS-Schlüssel.

9. (Optional) Geben Sie eine Beschreibung für den Replikatschlüssel ein.

   Die Konsole zeigt die aktuelle Beschreibung des Primärschlüssels an, Sie können sie jedoch ändern. Beschreibungen sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten dieselbe Beschreibung oder verschiedene Beschreibungen geben. AWS KMS synchronisiert die Beschreibungen von multiregionalen Schlüsseln nicht.

10. (Optional) Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wenn Sie mehrere Tags zum Replikatschlüssel zuweisen möchten, wählen Sie Add tag (Tag hinzufügen) aus.

    Die Konsole zeigt die Tags an, die aktuell mit dem Primärschlüssel verknüpft sind, Sie können sie jedoch ändern. Tags sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten dieselben Tags oder verschiedene Tags zuweisen. AWS KMS synchronisiert die Tags von multiregionalen Schlüsseln nicht.

    Wenn Sie einen KMS-Schlüssel markieren oder entmarkieren, wird dadurch möglicherweise die Berechtigung für den KMS-Schlüssel erteilt oder verweigert. Details dazu finden Sie unter ABAC für AWS KMS und Verwenden von Tags zur Steuerung des Zugriffs auf KMS-Schlüssel.

11. Wählen Sie die IAM-Benutzer und -Rollen aus, die den Replikatschlüssel verwalten können.

    > **Note**
    >
    > IAM-Richtlinien können anderen IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwaltung der Replikatschlüssel erteilen.
    > Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre

Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Dieser Schritt startet den Prozess zum Erstellen einer Schlüsselrichtlinie für den Replikatschlüssel. Die Konsole zeigt die aktuelle Schlüsselrichtlinie des Primärschlüssels an, Sie können sie jedoch ändern. Schlüsselrichtlinien sind keine gemeinsame Eigenschaft von multiregionalen Schlüsseln. Sie können Ihrem multiregionalen Primärschlüssel und seinen Replikaten die gleiche Schlüsselrichtlinie oder andere Schlüsselrichtlinien zuweisen. Von AWS KMS werden Schlüsselrichtlinien nicht synchronisiert. Sie können die Schlüsselrichtlinie eines jeden KMS-Schlüssels jederzeit ändern.

12. Führen Sie die Schritte zum Erstellen der Schlüsselrichtlinie durch, einschließlich der Auswahl von Schlüsselbenutzern. Wählen Sie nach dem Überprüfen der Schlüsselrichtlinie Finish (beenden) aus, um den Replikatschlüssel zu erstellen.

Erstellen eines Replikatschlüssels (AWS KMS-API)

Um einen multiregionalen Replikatschlüssel zu erstellen, verwenden Sie die -ReplicateKeyOperation. Sie können die -CreateKeyOperation nicht verwenden, um einen Replikatschlüssel zu erstellen. Bei dieser Produktion wird jeweils ein Replikatschlüssel erstellt. Die Region, die Sie angeben, muss den Anforderungen der Region für Replikatschlüssel entsprechen.

Wenn Sie die `ReplicateKey`-Produktion verwenden, müssen Sie keine Werte für gemeinsam genutzte Eigenschaften von multiregionalen Schlüsseln angeben. Gemeinsam genutzte Eigenschaftswerte werden aus dem Primärschlüssel kopiert und synchronisiert. Sie können jedoch Werte für nicht freigegebene Eigenschaften angeben. Ansonsten wendet AWS KMS die Standardwerte für KMS-Schlüssel an, nicht die Werte des Primärschlüssels.

> **Note**
>
> Wenn Sie keine Werte für die `Description`-, `KeyPolicy`-, oder `Tags`-Parameter angeben, erstellt AWS KMS den Replikatschlüssel mit einer leeren Zeichenfolgen-Beschreibung, der Standard-Schlüsselrichtlinie und ohne Tags.
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Beispielsweise erstellt der folgende Befehl einen multiregionalen Replikatschlüssel in der Region Asien-Pazifik (Sydney) (ap-southeast-2). Dieser Replikatschlüssel wird nach dem Primärschlüssel in der Region USA Ost (Nord-Virginia) (us-east-1) modelliert, der durch den Wert des `KeyId`-Parameters identifiziert wird. Dieses Beispiel akzeptiert Standardwerte für alle anderen Eigenschaften, einschließlich der Schlüsselrichtlinie.

Die Antwort beschreibt den neuen Replikatschlüssel. Sie enthält Felder für gemeinsam genutzte Eigenschaften, z. B. `KeyId`, `KeySpec` und `KeyUsage`, und den Ursprung des Schlüsselmaterials (`Origin`). Sie enthält auch Eigenschaften, die unabhängig vom Primärschlüssel sind, z. B. die `Description`, Schlüsselrichtlinie (`ReplicaKeyPolicy`) und Tags (`ReplicaTags`).

Die Antwort enthält auch den Schlüssel-ARN und die Region des Primärschlüssels und aller seiner Replikatschlüssel, einschließlich des Schlüssels, der gerade in der Region ap-southeast-2 erstellt wurde. In diesem Beispiel zeigt das `ReplicaKey`-Element an, dass dieser Primärschlüssel bereits in der Region Europa (Irland) (eu-west-1) repliziert wurde.

```
$ 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": []
}
```

Anzeigen von multiregionalen Schlüsseln

Sie können einzelregionale und multiregionale Schlüssel in der AWS KMS-Konsole und mithilfe der AWS KMSAPI-Operationen anzeigen.

Themen

- Anzeigen von multiregionalen Schlüsseln in der Konsole
- Anzeigen von multiregionalen Schlüsseln in der API

Anzeigen von multiregionalen Schlüsseln in der Konsole

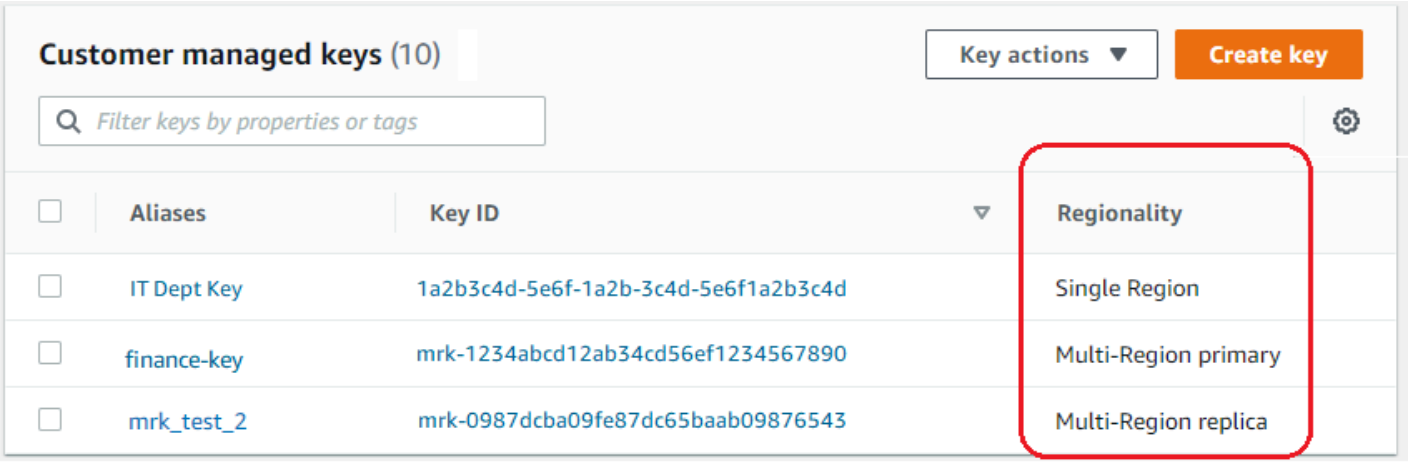
In der AWS KMS-Konsole können Sie KMS-Schlüssel in der ausgewählten Region anzeigen. Wenn Sie jedoch einen multiregionalen Schlüssel haben, können Sie die verwandten multiregionalen Schlüssel in anderen AWS-Regionen sehen.

Die Tabelle Customer managed keys (kundenverwaltete Schlüssel) in der AWS KMS-Konsole zeigt nur KMS-Schlüssel in der ausgewählten Region an. Sie können multiregionale Primär- und

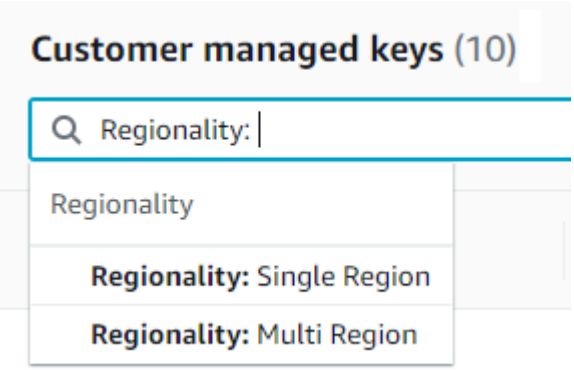
Replikatschlüssel in der ausgewählten Region anzeigen. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.

Die Von AWS verwaltete Schlüssel-Tabelle verfügt über keine Regionenfunktionen, da Von AWS verwaltete Schlüssel immer einzelregionale Schlüssel sind.

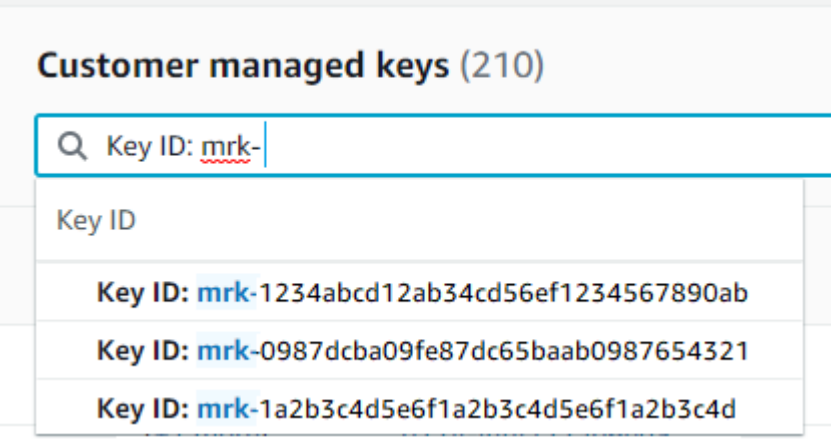
- Um die Identifizierung Ihrer multiregionalen Schlüssel zu erleichtern, fügen Sie die Spalte Regionality (Regionalität) in Ihre Schlüsseltabelle ein. Weitere Informationen dazu finden Sie unter Anpassen Ihrer KMS-Schlüsseltabellen.



- Um in der Schlüsseltabelle nur einzelregionale Schlüssel oder nur multiregionale Schlüssel anzuzeigen, filtern Sie Ihre Schlüssel nach der Regionalität-Eigenschaft jedes Schlüssels. Weitere Informationen dazu finden Sie unter Sortieren und Filtern Ihrer KMS-Schlüssel.



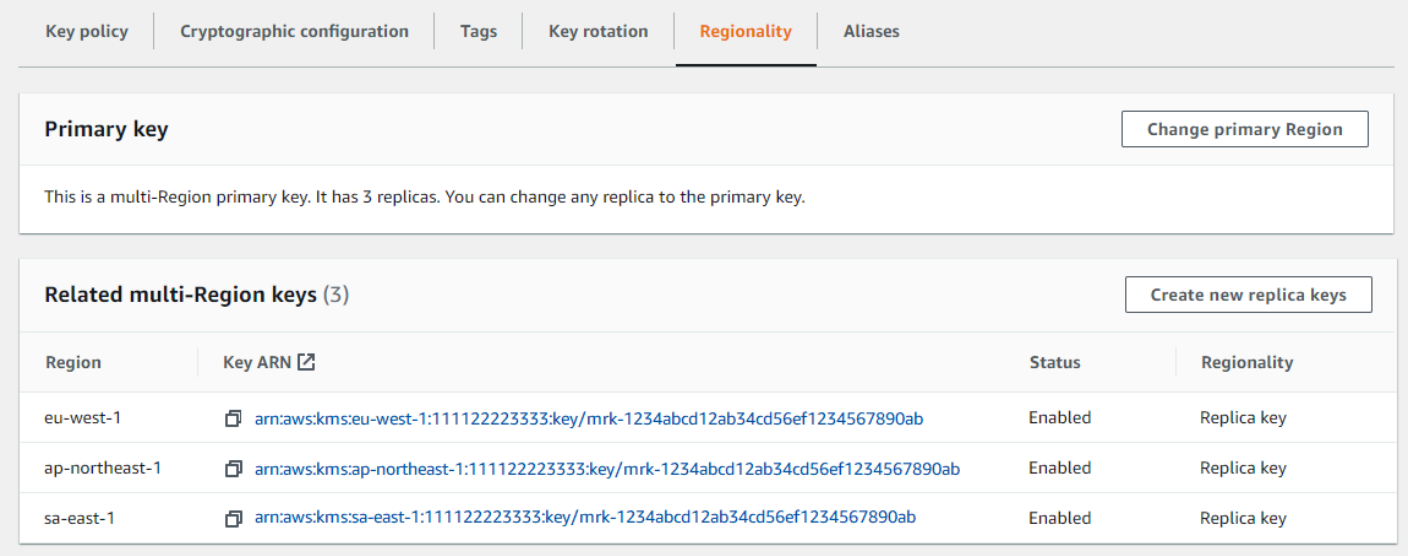
- Sie können auch die Tabelle Customer managed keys (kundenverwaltete Schlüssel) nach der unverwechselbaren mrk-Schlüssel-ID-Präfix sortieren und filtern.



- Weitere Informationen zu einem multiregionalen Primärschlüssel finden Sie auf der Detailseite für den Schlüssel unter der Registerkarte Regionality (Regionalität).

  Die Registerkarte Regionality (Regionalität) eines Primärschlüssels enthält die Schaltflächen Primärbereich ändern und Neue Replikatschlüssel erstellen. (Die Regionalität-Registerkarte für einen Replikatschlüssel enthält keine dieser Schaltflächen.) Der Abschnitt Related multi-Region keys (verwandte multiregionale Schlüssel) listet alle multiregionale Schlüssel auf, die mit dem aktuellen verwandt sind. Wenn der aktuelle Schlüssel ein Replikatschlüssel ist, enthält diese Liste den Primärschlüssel.

  Wenn Sie einen verwandten multiregionalen Schlüssel aus der Tabelle Related multi-Region keys (verwandte multiregionale Schlüssel) auswählen, wechselt die AWS KMS-Konsole in die Region des ausgewählten Schlüssels und öffnet die Detailseite für den Schlüssel. Wenn Sie beispielsweise den Replikatschlüssel in der `sa-east-1`-Region aus dem Beispielsabschnitt Related multi-Region keys (verwandte multiregionale Schlüssel) unten auswählen, wechselt dieAWS KMS-Konsole in die Region `sa-east-1`, um die Detailseite für diesen Replikatschlüssel anzuzeigen. Sie könnten dies tun, um den Alias- oder die Schlüsselrichtlinie für den Replikatschlüssel anzuzeigen. Um die Region erneut zu ändern, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.



Anzeigen von multiregionalen Schlüsseln in der API

Um multiregionale Schlüssel in der AWS KMS API anzuzeigen, verwenden Sie die -DescribeKeyOperation. Sie zeigt den angegebenen Schlüssel und alle zugehörigen multiregionalen Schlüssel an.

Wie die AWS KMS-Konsole, sind AWS KMS-API-Operationen regional. Wenn Sie beispielsweise die ListAliases Operationen ListKeys oder aufrufen, geben sie nur die Ressourcen in der aktuellen oder der angegebenen Region zurück. Aber wenn Sie die `DescribeKey`-Operation für einen multiregionalen Schlüssel aufrufen, enthält die Antwort alle verwandte multiregionale Schlüssel in anderen AWS-Regionen.

Im folgenden Beispiel erhält die `DescribeKey`-Anforderung Details zu einem multiregionalen Replikatschlüssel in der Region Asien-Pazifik (Tokio) (`ap-northeast-1`).

```
$ aws kms describe-key \
        --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
        --region ap-northeast-1
```

Die meisten `KeyMetadata` in der Antwort beschreiben den Replikatschlüssel in der Region Asien-Pazifik (Tokio), der Gegenstand der Anforderung ist. Allerdings beschreibt das `MultiRegionConfiguration`-Element den Primärschlüssel in der Region USA West (Oregon) (us-west-2) und seine Replikatschlüssel in anderen AWS-Regionen, einschließlich des Replikats in

der Region Asien-Pazifik (Tokio). `DescribeKey`gibt den gleichen `MultiRegionConfiguration`-Wert für alle verwandte multiregionale Schlüssel zurück.

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1586329200.918,
        "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"
```

```
                }
            ]
        }
    }
}
```

Verwalten von multiregionalen Schlüsseln

Verwenden und verwalten Sie für die meisten Aktionen die multiregionalen Schlüssel auf die gleiche Weise, wie Sie einzelregionale Schlüssel verwenden und verwalten. Sie können die Schlüssel aktivieren und deaktivieren, Aliasse, Schlüsselrichtlinien, Erteilungen und Tags festlegen und aktualisieren. Die Verwaltung von multiregionalen Schlüsseln unterscheidet sich jedoch folgendermaßen.

- Sie können die primäre Region aktualisieren. Dadurch wird einer der Replikatschlüssel in einen Primärschlüssel und der aktuelle Primärschlüssel in ein Replikat geändert.
- Sie verwalten die automatische Drehung nur für den Primärschlüssel.
- Sie können den öffentlichen Schlüssel für einen asymmetrischen multiregionalen Schlüssel aus einem der zugehörigen Primär- oder Replikatschlüssel erhalten.

Die multiregionale Eigenschaft, die Sie bei der Erstellung eines KMS-Schlüssels festlegen, ist unveränderlich. Einzelregionale Schlüssel können nicht in multiregionale Schlüssel konvertiert werden oder umgekehrt.

Aktualisieren der primären Region

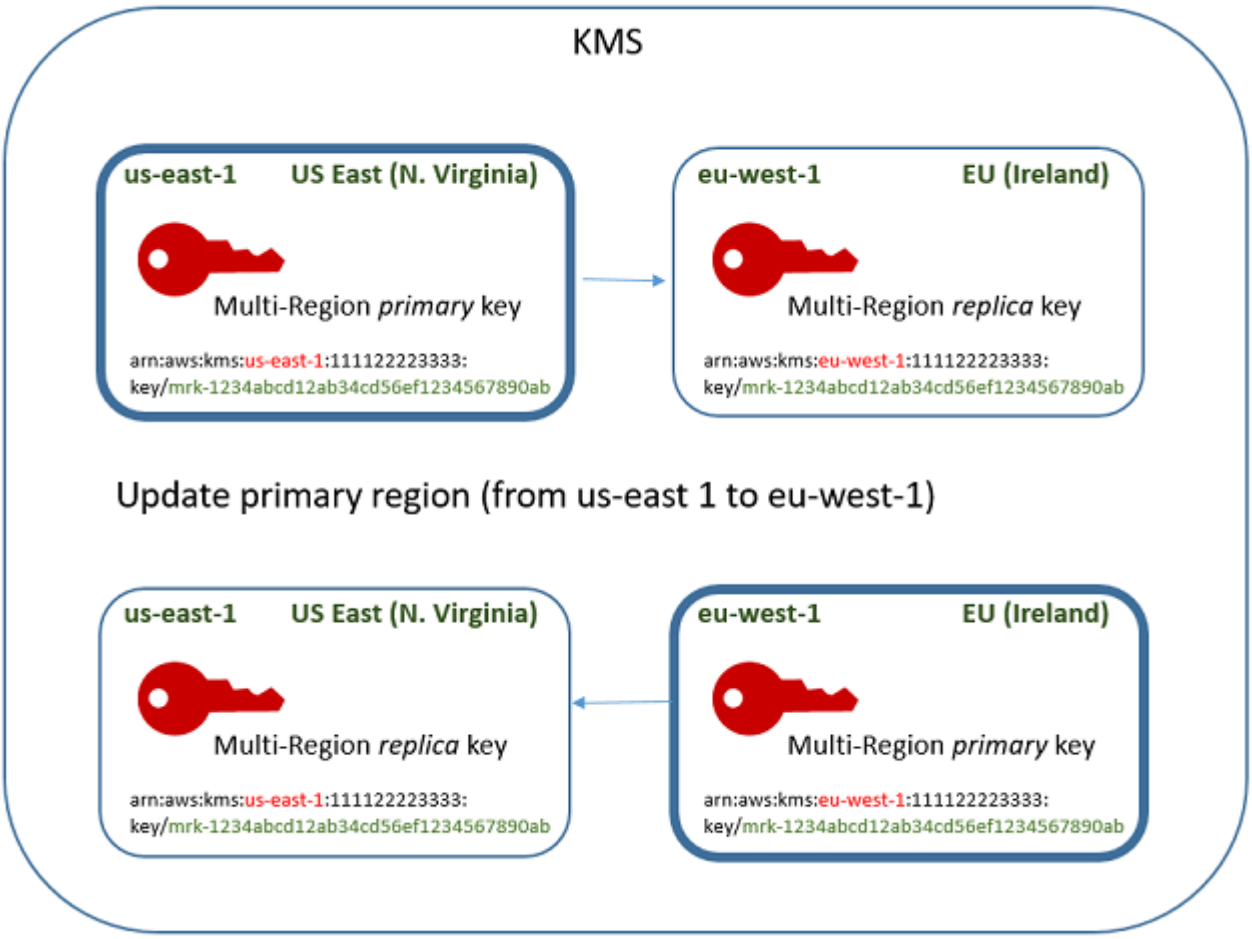
Jeder Satz verwandter multiregionaler Schlüssel muss über einen Primärschlüssel verfügen. Sie können jedoch den Primärschlüssel ändern. Diese Aktion, bekannt als Aktualisieren der primären Region, konvertiert den aktuellen Primärschlüssel in einen Replikatschlüssel und konvertiert einen der verwandten Replikatschlüssel in den Primärschlüssel. Sie können dies tun, wenn Sie den aktuellen Primärschlüssel löschen müssen, während die Replikatschlüssel beibehalten werden, oder um den Primärschlüssel in derselben Region wie die Schlüsseladministratoren zu platzieren.

Sie können einen beliebigen verwandten Replikatschlüssel als neuen Primärschlüssel auswählen. Sowohl der Primärschlüssel als auch der Replikatschlüssel müssen sich im Schlüsselstatus `Enabled` befinden, wenn die Operation gestartet wird.

Selbst nachdem diese Operation abgeschlossen ist, kann der Vorgang zum Aktualisieren der primären Region noch einige Sekunden dauern. Während dieser Zeit haben der alte und der neue

Primärschlüssel den vorübergehenden Schlüsselstatus Updating. Während der Schlüsselstatus lautet `Updating` ist, können Sie die Schlüssel in kryptografischen Vorgängen verwenden, aber Sie können den neuen Primärschlüssel nicht replizieren oder bestimmte Verwaltungsvorgänge ausführen, wie z. B. das Aktivieren oder Deaktivieren dieser Schlüssel. Operationen wie DescribeKeykönnten sowohl den alten als auch den neuen Primärschlüssel als Replikate anzeigen. Die `Enabled`-Schlüsselstatus wird wiederhergestellt, wenn die Aktualisierung abgeschlossen ist.

Angenommen, Sie haben einen Primärschlüssel in USA Ost (Nord-Virginia) (us-east-1) und einen Replikatschlüssel in Europa (Irland) (eu-west-1). Sie können die Aktualisierungsfunktion verwenden, um den Primärschlüssel in USA Ost (Nord-Virginia) (us-east-1) in einen Replikatschlüssel zu ändern und den Replikatschlüssel in Europa (Irland) (eu-west-1) in den Primärschlüssel zu ändern.



Wenn der Aktualisierungsvorgang abgeschlossen ist, ist der multiregionaler Schlüssel in der Region Europa (Irland) (eu-west-1) ein multiregionaler Primärschlüssel und der Schlüssel in der Region USA Ost (Nord-Virginia) (us-east-1) ist sein Replikatschlüssel. Wenn andere verwandte Replikatschlüssel vorhanden sind, werden sie zu Replikaten des neuen Primärschlüssels. Beim nächsten AWS KMS Synchronisieren der gemeinsamen Eigenschaften der Schlüssel mit mehreren Regionen werden

die gemeinsamen Eigenschaften aus dem neuen Primärschlüssel abgerufen und in die zugehörigen Replikatschlüssel kopiert, einschließlich des früheren Primärschlüssels.

Der Aktualisierungsvorgang hat keine Auswirkungen auf den Schüssel-ARN eines beliebigen multiregionalen Schlüssels. Es hat auch keine Auswirkungen auf gemeinsame Eigenschaften, wie das Schlüsselmaterial, oder auf unabhängige Eigenschaften, wie die Schlüsselrichtlinie. Sie können ggf. die Schlüsselrichtlinie des neuen Primärschlüssels aktualisieren. Sie könnten zum Beispiel dem neuen Primärschlüssel kms: ReplicateKey Permission for Trusted Principals hinzufügen und ihn aus dem neuen Replikatschlüssel entfernen.

Der **`Updating`**-Schlüsselstatus

Die Aktualisierung einer primären Region dauert etwas länger als die kurze eventuelle Konsistenzverzögerung, von der die meisten Vorgänge betroffen sind. AWS KMS Der Prozess wird möglicherweise noch ausgeführt, nachdem die `UpdatePrimaryRegion`-Operation zurückgibt, oder nachdem Sie den Aktualisierungsvorgang in der Konsole abgeschlossen haben. Bei Vorgängen wie werden DescribeKeymöglicherweise sowohl der alte als auch der neue Primärschlüssel als Replikate angezeigt, bis der Vorgang abgeschlossen ist.

Während der Aktualisierung der primären Region befinden sich der alte Primärschlüssel und der neue Primärschlüssel im `Updating`-Schlüsselstatus. Wenn der Aktualisierungsvorgang erfolgreich abgeschlossen ist, kehren beide Schlüssel zum `Enabled`-Schlüsselstatus zurück. Im `Updating`-Status sind einige Verwaltungsvorgänge, wie das Aktivieren und Deaktivieren der Schlüssel, nicht verfügbar. Sie können jedoch weiterhin beide Schlüssel ohne Unterbrechung in kryptografischen Operationen verwenden. Informationen zu den Auswirkungen des `Updating`-Schlüsselstatus finden Sie unter Wichtige Zustände von AWS KMS Schlüsseln.

Aktualisieren einer primären Region (Konsole)

Sie können den Primärschlüssel in der AWS KMS Konsole aktualisieren. Starten Sie auf der Seite mit den Schlüsseldetails des aktuellen Primärschlüssels.

1. Melden Sie sich bei der AWS Key Management Service (AWS KMS) -Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie die Schlüssel-ID oder den Alias des multiregionalen Primärschlüssels aus. Dadurch wird die Seite mit den Schlüsseldetails des Primärschlüssels geöffnet.

Um einen multiregionalen Primärschlüssel zu identifizieren, verwenden Sie das Werkzeugsymbol in der oberen rechten Ecke, um die Spalte Regionality (Regionalität) zur Tabelle hinzuzufügen.

5. Wählen Sie die Registerkarte Regionality (Regionalität) aus.
6. In dem Abschnitt Primary key (Primärschlüssel) wählen Sie Change primary Region (primäre Region ändern) aus.
7. Wählen Sie die Region des neuen Primärschlüssels aus. Sie können nur eine Region aus dem Menü auswählen.

   Das Menü Change primary Regions (primären Regionen ändern) enthält nur Regionen, die über einen verwandten multiregionalen Schlüssel verfügen. Möglicherweise haben Sie nicht die Berechtigung zum Aktualisieren der primären Region in allen Regionen auf dem Menü.
8. Wählen Sie Change primary region (Primäre Region ändern) aus.

Aktualisierung einer primären Region (API)AWS KMS

Verwenden Sie den UpdatePrimaryRegionVorgang, um den Primärschlüssel in einer Reihe verwandter Schlüssel für mehrere Regionen zu ändern.

Verwenden Sie den `KeyId`-Parameter, um den aktuellen Primärschlüssel zu identifizieren. Verwenden Sie den `PrimaryRegion` Parameter, um den AWS-Region des neuen Primärschlüssels anzugeben. Wenn der Primärschlüssel noch kein Replikat in der neuen primären Region hat, schlägt die Operation fehl.

Im folgenden Beispiel wird der Primärschlüssel aus dem multiregionalen Schlüssel in der `us-west-2`-Region zu seinem Replikat im `eu-west-1`-Region geändert. Der `KeyId`-Parameter identifiziert den aktuellen Primärschlüssel in der Region `us-west-2`. Der `PrimaryRegion` Parameter gibt den AWS-Region des neuen Primärschlüssels an,`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
```

Wenn diese Operation erfolgreich ist, gibt sie keine Ausgabe zurück, sondern nur den HTTP-Statuscode. Um den Effekt zu sehen, rufen Sie den DescribeKeyVorgang für einen der Multi-Region-Schlüssel auf. Möglicherweise möchten Sie warten, bis der Schlüsselstatus zu `Enabled` zurückkehrt.

Während der Schlüsselstatus Updating ist, befinden sich die Werte für den Schlüssel möglicherweise noch im Fluss.

Beispielsweise erhält der folgenden `DescribeKey`-Aufruf die Details über den multiregionalen Schlüssel in der `eu-west-1`-Region. Die Ausgabe zeigt, dass der multiregionale Schlüssel in der Region `eu-west-1` ist jetzt der Primärschlüssel. Der verwandte multiregionale Schlüssel (gleiche Schlüssel-ID) in der Region `us-west-2` ist jetzt ein Replikatschlüssel.

```
$ 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"
```

```
                }
            ]
        }
    }
}
```

Drehen von multiregionalen Schlüsseln

In Tasten für mehrere Regionen können Sie die automatische Rotation aktivieren und deaktivieren und bei Bedarf die Rotation des Schlüsselmaterials durchführen. Die Schlüsselrotation ist eine gemeinsame Eigenschaft von Schlüsseln mit mehreren Regionen.

Sie können die automatische Schlüsseldrehung nur für einen Primärschlüssel aktivieren oder deaktivieren. Sie initiieren die Rotation bei Bedarf nur für den Primärschlüssel.

- Bei der AWS KMS Synchronisierung der Schlüssel für mehrere Regionen wird die Eigenschaftseinstellung für die Schlüsselrotation vom Primärschlüssel auf alle zugehörigen Replikatschlüssel kopiert.
- Bei der AWS KMS Rotation des Schlüsselmaterials wird neues Schlüsselmaterial für den Primärschlüssel erstellt. Anschließend wird das neue Schlüsselmaterial über Regionsgrenzen hinweg in alle zugehörigen Replikatschlüssel kopiert. Das Schlüsselmaterial wird niemals unverschlüsselt verlassen AWS KMS . Dieser Schritt wird sorgfältig überwacht, um sicherzustellen, dass das Schlüsselmaterial vollständig synchronisiert wird, bevor ein Schlüssel in einer kryptografischen Operation verwendet wird.
- AWS KMS verschlüsselt keine Daten mit dem neuen Schlüsselmaterial, bis dieses Schlüsselmaterial im Primärschlüssel und in jedem seiner Replikatschlüssel verfügbar ist.
- Wenn Sie einen gedrehten Primärschlüssel replizieren, enthält der neue Replikatschlüssel das aktuelle Schlüsselmaterial und alle früheren Versionen des Schlüsselmaterials für seine verwandten multiregionalen Schlüssel.

Dieses Muster stellt sicher, dass verwandte multiregionale Schlüssel vollständig interoperabel sind. Jeder multiregionale Schlüssel kann Chiffretext entschlüsseln, der mit einem verwandten multiregionalen Schlüssel verschlüsselt wurde, selbst wenn der Chiffretext vor dem Erstellen des Schlüssels verschlüsselt wurde.

Automatische Schlüsseldrehung wird für asymmetrische KMS-Schlüssel oder KMS-Schlüssel mit importiertem Schlüsselmaterial nicht unterstützt. Informationen zur automatischen Schlüsselrotation und zur bedarfsgesteuerten Schlüsselrotation finden Sie unter. Rotierend AWS KMS keys

Herunterladen öffentlicher Schlüssel

Wenn Sie einen asymmetrischen KMS-Schlüssel mit mehreren Regionen erstellen, AWS KMS wird ein RSA- oder ECC-Schlüsselpaar (Elliptic Curve) für den Primärschlüssel erstellt. Dann kopiert es dieses Schlüsselpaar in jedes Replikat des Primärschlüssels. Daher können Sie den öffentlichen Schlüssel aus dem Primärschlüssel oder einem seiner Replikatschlüssel herunterladen. Sie erhalten immer das gleiche Schlüsselmaterial.

Hinweise zum Herunterladen und Verwenden von öffentlichen Schlüsseln außerhalb von finden Sie unter. AWS KMSBesondere Überlegungen zum Herunterladen öffentlicher Schlüssel Detaillierte Anweisungen finden Sie unter Herunterladen öffentlicher Schlüssel.

Schlüsselmaterial in multiregionale Schlüssel importieren

Sie können Ihr eigenes Schlüsselmaterial in einen multiregionalen KMS-Schlüssel importieren Die multiregionale Schlüssel, die Sie mit Ihrem eigenen Schlüsselmaterial erstellen, sind interoperabel. Sie können Daten in einer Region verschlüsseln und in einer beliebigen anderen Region mit einem verwandten multiregionalen Schlüssel entschlüsseln.

Sie müssen jedoch das Schlüsselmaterial verwalten.

- AWS KMS kopiert oder synchronisiert das Schlüsselmaterial aus einem Primärschlüssel mit importiertem Schlüsselmaterial nicht zu seinen Replikatschlüsseln. Sie müssen dasselbe Schlüsselmaterial in verwandte Primär- und Replikatschlüssel importieren.
- Sie legen das Ablaufmodell und das Ablaufdatum für jeden Schlüssel unabhängig voneinander fest, wenn Sie das Schlüsselmaterial importieren. Sie können dasselbe oder ein anderes Ablaufmodell und Ablaufdatum für verwandte multiregionale Schlüssel konfigurieren. Wenn sich das Schlüsselmaterial dem Ablaufdatum nähert, müssen Sie das Schlüsselmaterial erneut in den betroffenen multiregionalen Schlüssel importieren.

  Der Schlüsselstatus der verwandten multiregionalen Schlüssel ist von den anderen unabhängig. Wenn beispielsweise das Schlüsselmaterial im Primärschlüssel abläuft, bleiben die Replikatschlüssel davon unberührt.

Die gleichen Regionsanforderungen für Replikatschlüssel gelten für multiregionale Schlüssel mit importiertem Schlüsselmaterial. Wenn Sie dasselbe Schlüsselmaterial in einzelregionale Schlüssel oder nicht verwandte multiregionale Schlüssel importieren, sind diese KMS-Schlüssel nicht interoperabel.

Sie können multiregionale Schlüssel mit importiertem symmetrischem, asymmetrischem oder HMAC-Schlüsselmaterial erstellen. AWS KMS unterstützt importiertes Schlüsselmaterial in benutzerdefinierten Schlüsselspeichern nicht. Außerdem ist es nicht möglich, die automatische Schlüsseldrehung für einen KMS-Schlüssel mit importiertem Schlüsselmaterial zu aktivieren.

Neben ihren multiregionalen Funktionen sind multiregionale Schlüssel mit importiertem Schlüsselmaterial identisch mit anderen KMS-Schlüsseln mit importiertem Schlüsselmaterial. Ausführliche Informationen zum Erstellen und Konfigurieren von einzelregionalen Schlüsseln mit importiertem Schlüsselmaterial finden Sie unter Informationen zu importiertem Schlüsselmaterial.

Themen

Warum sind nicht alle KMS-Schlüssel mit importiertem Schlüsselmaterial interoperabel?

Einzelregionale KMS-Schlüssel mit importiertem Schlüsselmaterial sind nicht interoperabel, selbst wenn sie dasselbe Schlüsselmaterial haben. Wenn AWS KMS einen KMS-Schlüssel zum Verschlüsseln von Daten verwendet, bindet es einige Schlüsselmetadaten kryptografisch an den Chiffretext. Dadurch wird der Chiffretext gesichert, so dass nur der KMS-Schlüssel, der die Daten verschlüsselt hat, diese Daten entschlüsseln kann.

Multiregionale Schlüssel sind so konzipiert, dass sie interoperabel sind. Neben dem gleichen Schlüsselmaterial haben sie dieselbe Schlüssel-ID und andere Metadaten. Daher können die Chiffretexte, die sie erzeugen, durch einen beliebigen verwandten multiregionalen Schlüssel entschlüsselt werden. Daher unterscheiden sich die Vertrauenseigenschaften von multiregionalen Schlüsseln und von einzelregionalen Schlüsseln. Für einige Kunden überwiegt der Vorteil der Entschlüsselung in mehreren Regionen jedoch den Sicherheitswert eines Chiffretextes, der von einem einzelnen KMS-Schlüssel in einer einzelnen AWS-Region abhängig ist.

Erstellen eines Primärschlüssels mit importiertem Schlüsselmaterial

Um einen Primärschlüssel mit importiertem Schlüsselmaterial zu erstellen, erstellen Sie zunächst einen KMS-Schlüssel ohne Schlüsselmaterial. Wenn Sie den Primärschlüssel ohne Schlüsselmaterial erstellen, müssen Sie die Schlüsselspezifikation angeben, die dem Typ des Schlüsselmaterials

entspricht, das Sie importieren möchten. Anschließend importieren Sie Ihr Schlüsselmaterial in den Primärschlüssel.

Das Verfahren zum Erstellen eines multiregionalen Primärschlüssels ohne Schlüsselmaterial entspricht fast dem Verfahren zum Erstellen eines einzelregionalen Schlüssels ohne Schlüsselmaterial. Der einzige Unterschied besteht darin, dass Sie angeben, dass es sich bei dem Schlüssel um einen multiregionalen Schlüssel handelt.

Die Berechtigungen für die Erstellung eines multiregionalen Primärschlüssels mit importiertem Schlüsselmaterial sind dieselben wie für die Erstellung eines multiregionalen Primärschlüssels mit Schlüsselmaterial, einschließlich der kms:- und iam:-Berechtigungen in einer IAM-Richtlinie. AWS KMS CreateKey CreateServiceLinkedRole Sie können die Bedingungsschlüssel kms:MultiRegionKeyType und kms:KeyOrigin verwenden, um die Berechtigung zum Erstellen von multiregionalen Primärschlüsseln mit importiertem Schlüsselmaterial zu gewähren oder zu verweigern.

Verwenden Sie beim Erstellen eines Primärschlüssels mit importiertem Schlüsselmaterial in der AWS KMS-Konsole die Einstellungen im Abschnitt Advanced options (Erweiterte Optionen). Diese Eigenschaften können nicht geändert werden, nachdem der KMS-Schlüssel erstellt wurde.

- Stellen Sie die Key material origin (Herkunft des Schlüsselmaterials) auf External (Import key material) (Extern (Schlüsselmaterial importieren)) ein.
- Unter Multi-Region replication (multiregionale Replikation), wählen Sie Allow this key to be replicated into other Regions (Replizieren dieses Schlüssels in andere Regionen zulassen) aus.

Wenn Sie die -CreateKeyOperation verwenden, um einen Primärschlüssel mit importiertem Schlüsselmaterial zu erstellen, verwenden Sie die `MultiRegion` Parameter `Origin` und und geben Sie die Parameter `KeySpec` und an`KeyUsage`. Im folgenden Beispiel wird ein `EXTERNAL`-KMS-Schlüssel erstellt, mit dem `ECC_NIST_P384`-Schlüsselmaterial importiert werden kann.

```
$ aws kms create-key --origin EXTERNAL --key-spec ECC_NIST_P384 --key-usage SIGN_VERIFY
 --multi-region
```

Das Ergebnis ist ein multiregionaler Primärschlüssel ohne Schlüsselmaterial und einem Schlüsselstatus von `PendingImport`.

Um diesen KMS-Schlüssel zu aktivieren, müssen Sie einen öffentlichen Schlüssel und ein Import-Token herunterladen, den öffentlichen Schlüssel verwenden, um Ihr Schlüsselmaterial zu

verschlüsseln und anschließend Ihr Schlüsselmaterial importieren. Anweisungen finden Sie unter Schlüsselmaterial für AWS KMS Schlüssel importieren.

Erstellen eines Replikatschlüssels mit importiertem Schlüsselmaterial

Sie können in der AWS KMS-Konsole einen multiregionalen Replikatschlüssel erstellen, oder mithilfe der AWS KMS-API-Operationen. Um einen multiregionalen Primärschlüssel mit importiertem Schlüsselmaterial zu replizieren, verwenden Sie dasselbe Verfahren, wie zum Erstellen eines Replikatschlüssels mit AWS KMS-Schlüsselmaterial. Das Ergebnis ist jedoch anders. Anstatt einen Replikatschlüssel mit demselben Schlüsselmaterial wie der Primärschlüssel zurückzugeben, gibt der Replikationsprozess einen Replikatschlüssel ohne Schlüsselmaterial und einen Schlüsselstatus von `PendingImport` zurück. Um den Replikatschlüssel zu aktivieren, müssen Sie dasselbe Schlüsselmaterial in den Replikatschlüssel importieren, den Sie in den Primärschlüssel importiert haben.

Obwohl es das Schlüsselmaterial nicht repliziert, erstellt AWS KMS den Replikatschlüssel mit der gleichen Schlüssel-ID, Schlüsselspezifikation, Schlüsselnutzung und Ursprung des Schlüsselmaterials wie beim Primärschlüssel. Außerdem wird sichergestellt, dass das Schlüsselmaterial, das Sie in den Replikatschlüssel importieren, mit dem Schlüsselmaterial identisch ist, das Sie in den Primärschlüssel importiert haben.

Einen multiregionalen Replikatschlüssel mit importiertem Schlüsselmaterial erstellen:

1. Erstellen Sie einen multiregionalen Primärschlüssel mit importiertem Schlüsselmaterial.
2. Führen Sie eine der folgenden Aufgaben aus.

   In der AWS KMS-Konsole wählen Sie einen multiregionalen Primärschlüssel mit importiertem Schlüsselmaterial aus. Wählen Sie danach auf der Registerkarte Regionality (Regionalität) Create new replica keys (Erstellen neuer Replikatschlüssel) aus. Anweisungen finden Sie unter Erstellen von Replikatschlüsseln (Konsole).

   Oder verwenden Sie die -ReplicateKeyOperation. Für den `KeyId`-Parameter geben Sie die Schlüssel-ID oder den Schlüssel-ARN eines multiregionalen Primärschlüssels und importiertem Schlüsselmaterial ein. Anweisungen finden Sie unter Erstellen eines Replikatschlüssels (AWS KMS-API).
3. Befolgen Sie für jeden neuen Replikatschlüssel die Schritte zum Herunterladen eines öffentlichen Schlüssels und Import-Tokens. Verwenden Sie den öffentlichen Schlüssel, um das Schlüsselmaterial des Primärschlüssels zu verschlüsseln, und importieren Sie dann das

Schlüsselmaterial des Primärschlüssels in den Replikatschlüssel. Sie benötigen einen anderen öffentlichen Schlüssel und ein Import-Token für jeden Replikatschlüssel.

Wenn das Schlüsselmaterial, das Sie versuchen, in den Replikatschlüssel zu importieren, nicht mit dem Schlüsselmaterial wie dem Primärschlüssel identisch ist, schlägt der Vorgang fehl. AWS KMS erfordert nicht, dass das Ablaufmodell und das Ablaufdatum koordiniert werden. Sie können jedoch Geschäftsregeln für Ihre multiregionale Schlüssel festlegen. Anweisungen finden Sie unter Schlüsselmaterial für AWS KMS Schlüssel importieren.

Berechtigungen zum Replizieren von Schlüsseln mit importierten Schlüsselmaterialien

Um einen Replikatschlüssel mit importiertem Schlüsselmaterial zu erstellen, müssen Sie über die folgenden Berechtigungen verfügen.

In der Region des Primärschlüssels:

- kms:ReplicateKey auf dem Primärschlüssel (in der Region des Primärschlüssels). Fügen Sie diese Berechtigung in die Schlüsselrichtlinie des Primärschlüssels oder in eine IAM-Richtlinie ein.

In der Region des Replikatschlüssels:

- kms:CreateKey in einer IAM-Richtlinie.
- kms:GetParametersForImport. Sie können diese Berechtigung in die Schlüsselrichtlinie des Replikatschlüssels oder in eine IAM-Richtlinie aufnehmen.
- kms:ImportKeyMaterial. Sie können diese Berechtigung in die Schlüsselrichtlinie des Replikatschlüssels oder in eine IAM-Richtlinie aufnehmen.
- kms:TagResource ist erforderlich, um Tags beim Replizieren zuzuweisen. Fügen Sie diese Berechtigung in eine IAM-Richtlinie in der Replikatregion ein.
- kms:CreateAlias ist erforderlich, um einen Schlüssel in der AWS KMS Konsole zu replizieren. Details hierzu finden Sie unter Steuern des Zugriffs auf Aliasse.

Löschen von multiregionalen Schlüsseln

Wenn Sie keinen multiregionalen Primär- oder Replikatschlüssel mehr verwenden, können Sie dessen Löschung planen.

Obwohl das Löschen von KMS-Schlüsseln immer mit Vorsicht erfolgen sollte, ist das Löschen eines Replikats eines multiregionalen Schlüssels weniger riskant, vorausgesetzt, dass der Primärschlüssel noch in AWS KMS existiert. Wenn Sie einen Replikatschlüssel aus seiner Region löschen, aber Chiffretext ermitteln, der unter dem gelöschten Schlüssel verschlüsselt wurde, können Sie diesen Chiffretext mit einem beliebigen verwandten multiregionalen Schlüssel entschlüsseln. Sie können den Replikatschlüssel auch neu erstellen, indem Sie den Primärschlüssel erneut in die Region des Replikatschlüssels replizieren.

Das Löschen eines Primärschlüssels und aller seiner Replikatschlüssel ist jedoch eine sehr gefährliche Produktion, die dem Löschen eines einzelregionalen Schlüssels entspricht.

> **Warning**
>
> Das Löschen eines KMS-Schlüssels ist ein endgültiger und potenziell gefährlicher Vorgang. Fahren Sie nur fort, wenn Sie sicher sind, dass Sie den KMS-Schlüssel später nicht mehr verwenden müssen. Wenn Sie nicht sicher sind, sollten Sie den KMS-Schlüssel deaktivieren, anstatt ihn zu löschen.

Um einen Primärschlüssel zu löschen, müssen Sie zunächst alle seine Replikatschlüssel löschen. Wenn Sie einen Primärschlüssel aus einer bestimmten Region löschen müssen, ohne dessen Replikatschlüssel zu löschen, ändern Sie den Primärschlüssel in einen Replikatschlüssel, indem Sie die primäre Region aktualisieren.

Bevor Sie das Löschen eines KMS-Schlüssels planen, lesen Sie die Hinweise im Löschen von AWS KMS keys Thema und die Themen, die erklären, wie Sie die frühere Verwendung eines KMS-Schlüssels ermitteln und wie Sie einen CloudWatch Alarm einrichten, der Sie während der Wartezeit über die Verwendung des KMS-Schlüssels informiert. Bevor Sie den Primärschlüssel eines asymmetrischen multiregionalen Schlüssel löschen, lesen Sie das Thema Löschen asymmetrischer Schlüssel.

Themen

- Berechtigungen zum Löschen von multiregionalen Schlüsseln
- Löschen eines Replikatschlüssels
- Löschen eines Primärschlüssels

Berechtigungen zum Löschen von multiregionalen Schlüsseln

Um das Löschen eines multiregionalen Schlüssels zu planen, benötigen Sie nur die folgende Berechtigung.

- kms:ScheduleKeyDeletion – um das Löschen des multiregionalen Schlüssels zu planen und seine Wartezeit festzulegen.

Es wird auch dringend empfohlen, dass Sie über die folgenden verwandten Berechtigungen verfügen.

- kms:CancelKeyDeletion – , um das geplante Löschen des multiregionalen Schlüssels abzubrechen.
- kms:DescribeKey – um den Schlüsselstatus des multiregionalen Schlüssels und die Liste der verwandten multiregionalen Schlüssel anzuzeigen.
- kms:DisableKey – um Ihnen die Möglichkeit zu geben, einen multiregionalen Schlüssel zu deaktivieren, anstatt ihn zu löschen.
- kms:EnableKey – , um die Funktionalität eines multiregionalen Schlüssels nach dem Abbrechen des Löschens wiederherzustellen.

Sie können auch die Berechtigung zum Replizieren des Primärschlüssels und zum Ändern des Primärschlüssels einschließen.

- kms:ReplicateKey
- kms:UpdateReplicaRegion

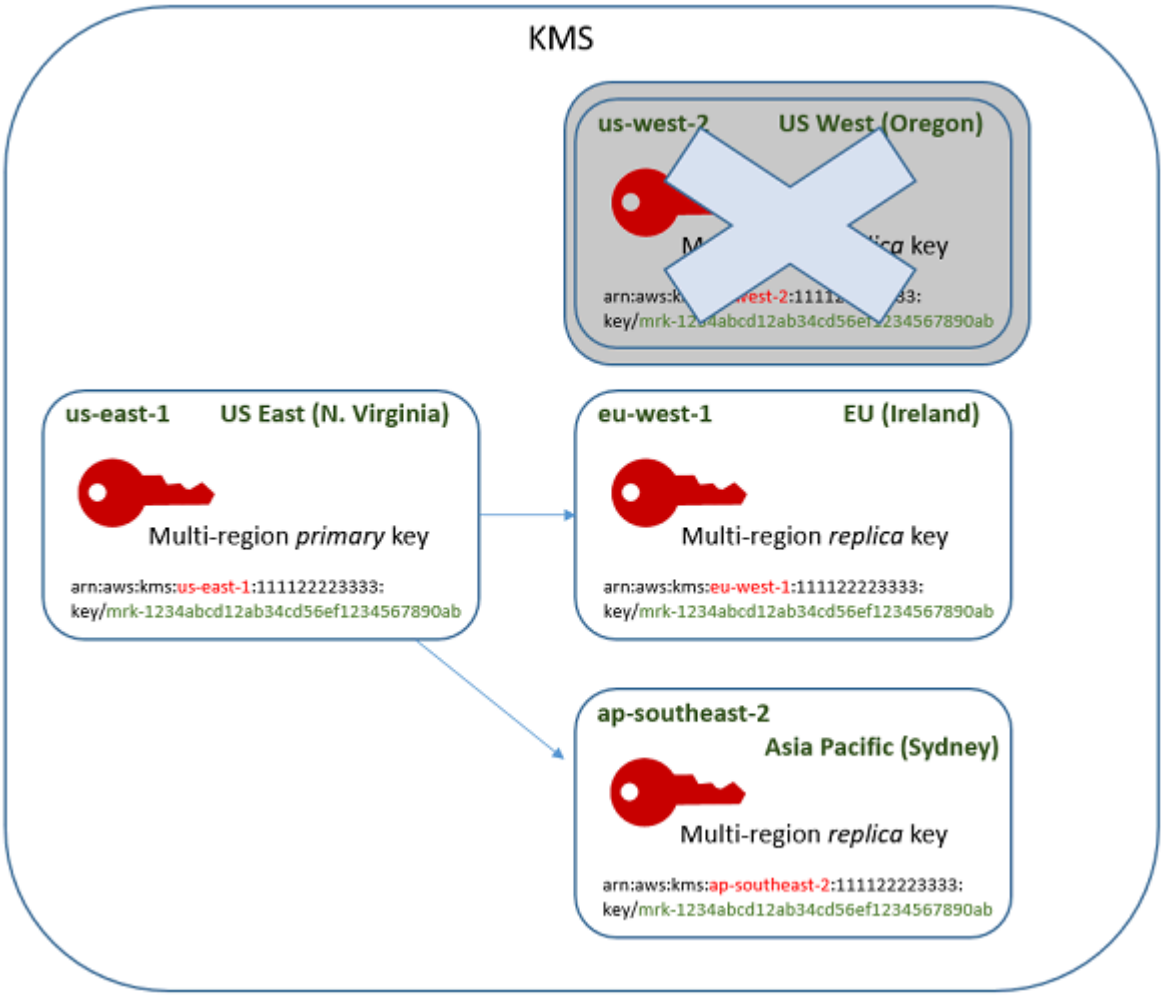
Sie können diese Berechtigungen in eine IAM-Richtlinie einschließen. Es ist jedoch eine bewährte Methode, sie in eine Schlüsselrichtlinie einzuführen, in der sie nur für den KMS-Schlüssel gelten, den Sie verwalten müssen.

Löschen eines Replikatschlüssels

Sie können die AWS KMS-Konsole oder die AWS KMS-API verwenden, um einen Replikatschlüssel zu löschen. Sie können einen Replikatschlüssel jederzeit löschen. Es hängt nicht vom Schlüsselstatus eines anderen KMS-Schlüssels ab.

Wenn Sie versehentlich einen Replikatschlüssel löschen, können Sie ihn neu erstellen, indem Sie denselben Primärschlüssel in derselben Region replizieren. Der neue Replikatschlüssel, den Sie erstellen, hat die gleichen gemeinsamen Eigenschaften wie der ursprüngliche Replikatschlüssel.

Das Verfahren zum Löschen eines multiregionalen Replikatschlüssels entspricht dem Löschen eines einzelregionalen Schlüssels.



1. Löschen des Replikatschlüssels planen. Wählen Sie eine Wartezeit von 7–30 Tagen. Die Standardwartezeit beträgt 30 Tage.
2. Während der Wartezeit ändert sich der Schlüsselstatus des Replikatschlüssels in `Pending deletion` (`PendingDeletion`) und Sie können ihn nicht in kryptografischen Produktionen verwenden.
3. Sie können das geplante Löschen des Replikatschlüssels zu einem beliebigen Zeitpunkt in der Wartezeit abbrechen. Der Schlüsselstatus wechselt zu `Disabled`, aber Sie können den KMS-Schlüssel erneut aktivieren.

4. Wenn die Wartezeit abgelaufen ist, löscht AWS KMS den Replikatschlüssel.

Sie können einen Datensatz Ihrer Aktionen in Ihrem AWS CloudTrail-Protokoll anzeigen. AWS KMS zeichnet die Produktionen auf, die das Löschen des KMS-Schlüssels planen und die Aktion, die den KMS-Schlüssel löscht.

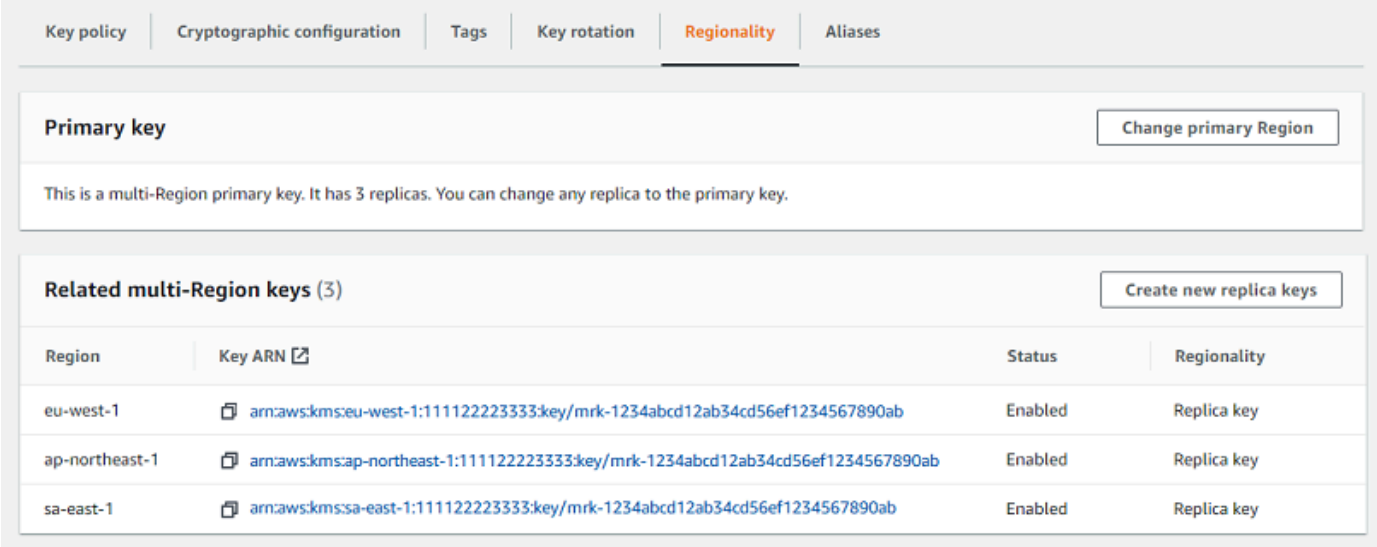
Löschen eines Replikatschlüssels (Konsole)

Um das Löschen eines multiregionalen Replikatschlüssels zu planen, verwenden Sie das gleiche Verfahren, mit dem Sie das Löschen eines einzelregionalen Schlüssels planen.

Da verwandte Replikatschlüssel in verschiedenen AWS-Regionen sind, können Sie nicht das Löschen von mehr als einem Replikatschlüssel gleichzeitig planen. Um alle verwandte Replikatschlüssel zu löschen, verwenden Sie ein Muster wie das folgende.

So planen Sie das Löschen aller verwandten Replikatschlüssel

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
3. Wählen Sie mit der Regionsauswahl in der oberen rechten Ecke die Region des multiregionalen Primärschlüssels aus.
4. Wählen Sie den Alias oder die Schlüssel-ID des Primärschlüssels aus.
5. Wählen Sie die Registerkarte Regionality (Regionalität) aus.



6. Unter Related multi-Region keys (verwandte multiregionale Schlüssel) wählen Sie die Schlüssel-ARN eines Replikatschlüssels aus.

   Diese Aktion öffnet die Seite mit den Schlüsseldetails des Replikatschlüssels in einer neuen Browser-Registerkarte. Die Konsole ist auf die Region des Replikatschlüssels festgelegt.

7. Wählen Sie im Menü Key actions (Schlüsselaktionen) Schedule key deletion (Schlüssellöschung planen) aus.

   Diese Aktion startet den Vorgang zur Planung der Löschung des Schlüssels. Den Vorgang zur Planung der Löschung des Schlüssels abschließen. Details hierzu finden Sie unter Planen und Abbrechen der Löschung eines Schlüssels (Konsole).

8. Kehren Sie zur Browser-Registerkarte zurück, auf der die Registerkarte Regionality (Regionalität) des Primärschlüssels angezeigt wird. (Möglicherweise müssen Sie die Seite aktualisieren, um den aktualisierten Status der Replikatschlüssel anzuzeigen. Wählen Sie den Schlüssel-ARN eines anderen Replikatschlüssels aus, und wiederholen Sie den Vorgang zur Planung der Löschung des Replikatschlüssels.

Löschen eines Replikatschlüssels (AWS KMS-API)

Um das Löschen eines multiregionalen Replikatschlüssels zu planen, verwenden Sie die -ScheduleKeyDeletionOperation. Um den KMS-Schlüssel anzugeben, verwenden Sie dessen Schlüssel-ID oder Schüssel-ARN. Wenn Sie mit multiregionalen Schlüsseln arbeiten, können Sie die Häufigkeit von Fehlern reduzieren, indem Sie den Schlüssel-ARN mit dem expliziten Regions-Wert verwenden.

Dieser Befehl löscht beispielsweise einen Replikatschlüssel aus der Region USA West (Oregon) (us-west-2). Da der Befehl keine Wartezeit angibt, wird die Wartezeit auf den Standardwert von 30 Tagen festgelegt.

```
$ aws kms schedule-key-deletion \
    --region us-west-2 \
    --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Wenn der Befehl erfolgreich ausgeführt wurde, werden der Schlüssel-ARN (`KeyId`), die Wartezeit (`PendingWindowInDays`) und das Löschdatum (`DeletionDate`) zurückgegeben, sowie der aktuelle Schlüsselstatus (`KeyState`), der voraussichtlich `PendingDeletion` sein wird.

Stellen Sie beim Löschen eines multiregionalen Replikatschlüssels sicher, dass die Schlüssel-ID und die Regions-Werte im Schlüssel-ARN die erwarteten Werte sind.

```
{
    "KeyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "DeletionDate": 1599523200.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 30
}
```

Um alle Replikate eines multiregionalen Primärschlüssels programmgesteuert zu löschen, erstellen Sie eine Liste der Regionen, die Replikatschlüssel enthalten. Rufen Sie dann für jede Region in der Liste die `ScheduleKeyDeletion`-Produktion auf, wie oben gezeigt.

Im Gegensatz zu einem einzelregionalen Schlüssel, der dauerhaft gelöscht wird, können Sie einen Replikatschlüssel wiederherstellen. Dazu replizieren Sie den Primärschlüssel in die Region, in der sich der gelöschte Replikatschlüssel befand.

Um den Status des Replikatschlüssels zu überprüfen und den Primärschlüssel und die Replikatschlüssel eines multiregionalen Schlüssels anzuzeigen, verwenden Sie die -DescribeKeyOperation.

Löschen eines Primärschlüssels

Sie können jederzeit das Löschen eines multiregionalen Primärschlüssels planen. AWS KMS löscht jedoch keinen multiregionalen Primärschlüssel, der Replikatschlüssel enthält, selbst wenn sie zum Löschen geplant sind.

Um einen Primärschlüssel zu löschen, müssen Sie das Löschen all seiner Replikatschlüssel planen und dann warten, bis die Replikatschlüssel gelöscht werden. Die erforderliche Wartezeit für das Löschen eines Primärschlüssels beginnt, wenn der letzte seiner Replikatschlüssel gelöscht wird. Wenn Sie einen Primärschlüssel aus einer bestimmten Region löschen müssen, ohne dessen Replikatschlüssel zu löschen, ändern Sie den Primärschlüssel in einen Replikatschlüssel, indem Sie die primäre Region aktualisieren.

Wenn ein Primärschlüssel keine Replikatschlüssel hat, ist der Prozess identisch mit dem Löschen eines Replikatschlüssels oder dem Löschen eines beliebigen regionalen KMS-Schlüssels.

Während ein Primärschlüssel zum Löschen geplant ist, können Sie ihn nicht in kryptografischen Produktionen verwenden, und Sie können ihn nicht replizieren. Die Replikatschlüssel sind jedoch nicht davon betroffen, es sei denn, sie werden ebenfalls zum Löschen geplant.

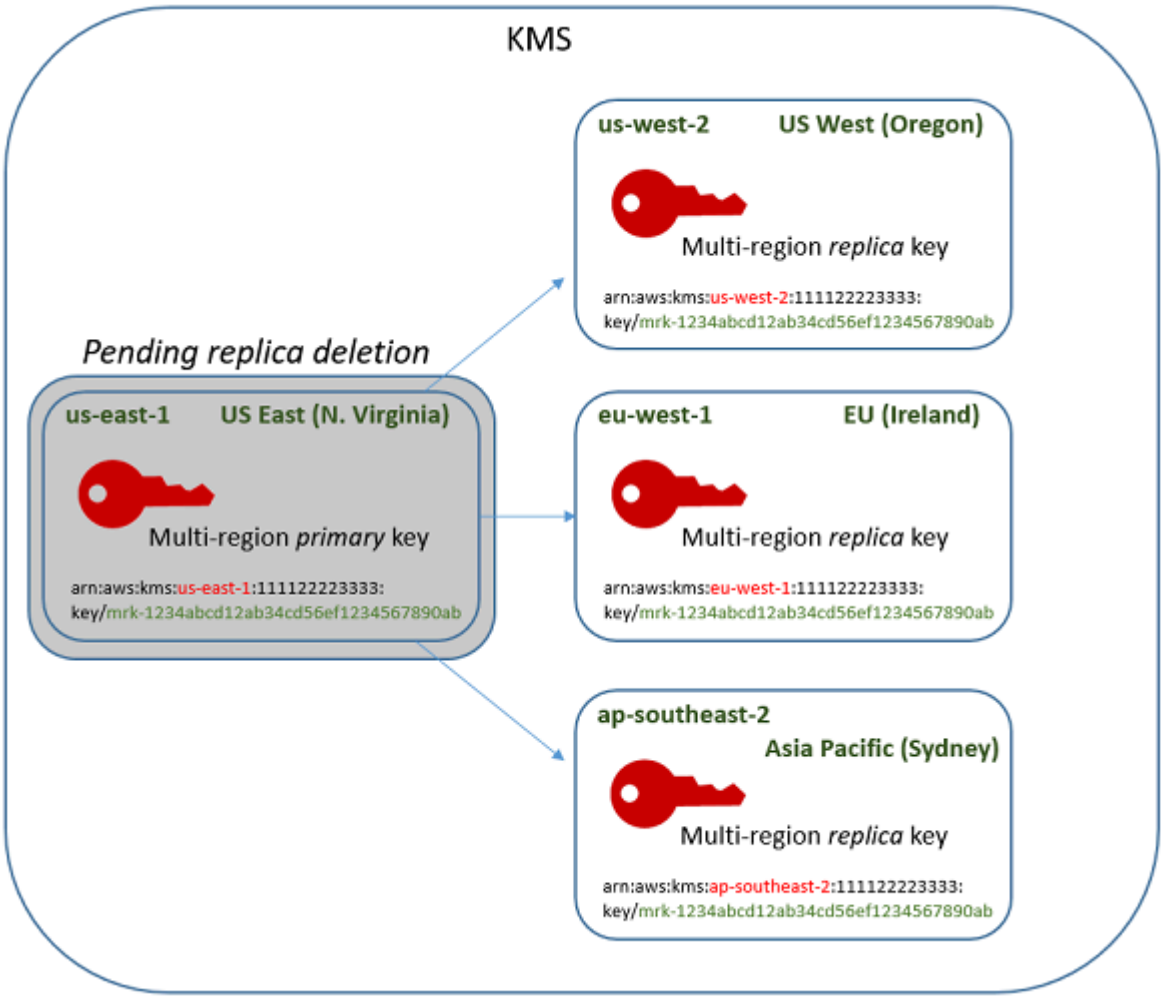
Sie können die AWS KMS-Konsole oder die AWS KMS-API verwenden, um das Löschen von Primärschlüsseln und Replikatschlüsseln zu planen. Sie können das Löschen des Primärschlüssels vor, nach oder gleichzeitig mit dem Löschen der Replikatschlüssel planen. Der Vorgang könnte in etwa wie folgt aussehen:

1. Planen Sie das Löschen des Primärschlüssels. Wählen Sie eine Wartezeit von 7–30 Tagen. Die Standardwartezeit beträgt 30 Tage. Die Wartezeit für den Primärschlüssel beginnt jedoch erst, wenn alle Replikatschlüssel gelöscht wurden.

   Wenn Replikatschlüssel noch vorhanden sind, wechselt der Schlüsselstatus des Primärschlüssels zu `Pending replica deletion` (`PendingReplicaDeletion`). Andernfalls ändert er sich in `Pending deletion` (`PendingDeletion`). In beiden Fällen können Sie den Primärschlüssel nicht in kryptografischen Produktionen verwenden, und Sie können ihn nicht replizieren.

   Das Planen des Löschens eines Primärschlüssels wirkt sich nicht auf die Replikatschlüssel aus. Ihr Schlüsselstatus bleibt aktiviert und Sie können sie in kryptografischen Produktionen verwenden. Wenn die Replikatschlüssel nicht gelöscht werden, kann der `Pending replica deletion`-Zustand des Primärschlüssels unbegrenzt beibehalten werden.

```
KMS key:                    Key state:
Primary (us-east-1)           Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Enabled
Replica (eu-west-1)           Enabled
Replica (ap-southeast-2)      Enabled
```



2. Planen Sie das Löschen jedes Replikatschlüssels. Wählen Sie eine Wartezeit von 7–30 Tagen. Die Standardwartezeit beträgt 30 Tage. Sie können mehrere Replikatschlüssel gleichzeitig löschen. Ihre Wartezeiten laufen gleichzeitig. Während der Wartezeit ändert sich der Schlüsselstatus der Replikatschlüssel in `Pending deletion` (`PendingDeletion`) und Sie können diese KMS-Schlüssel nicht in kryptografischen Produktionen verwenden.

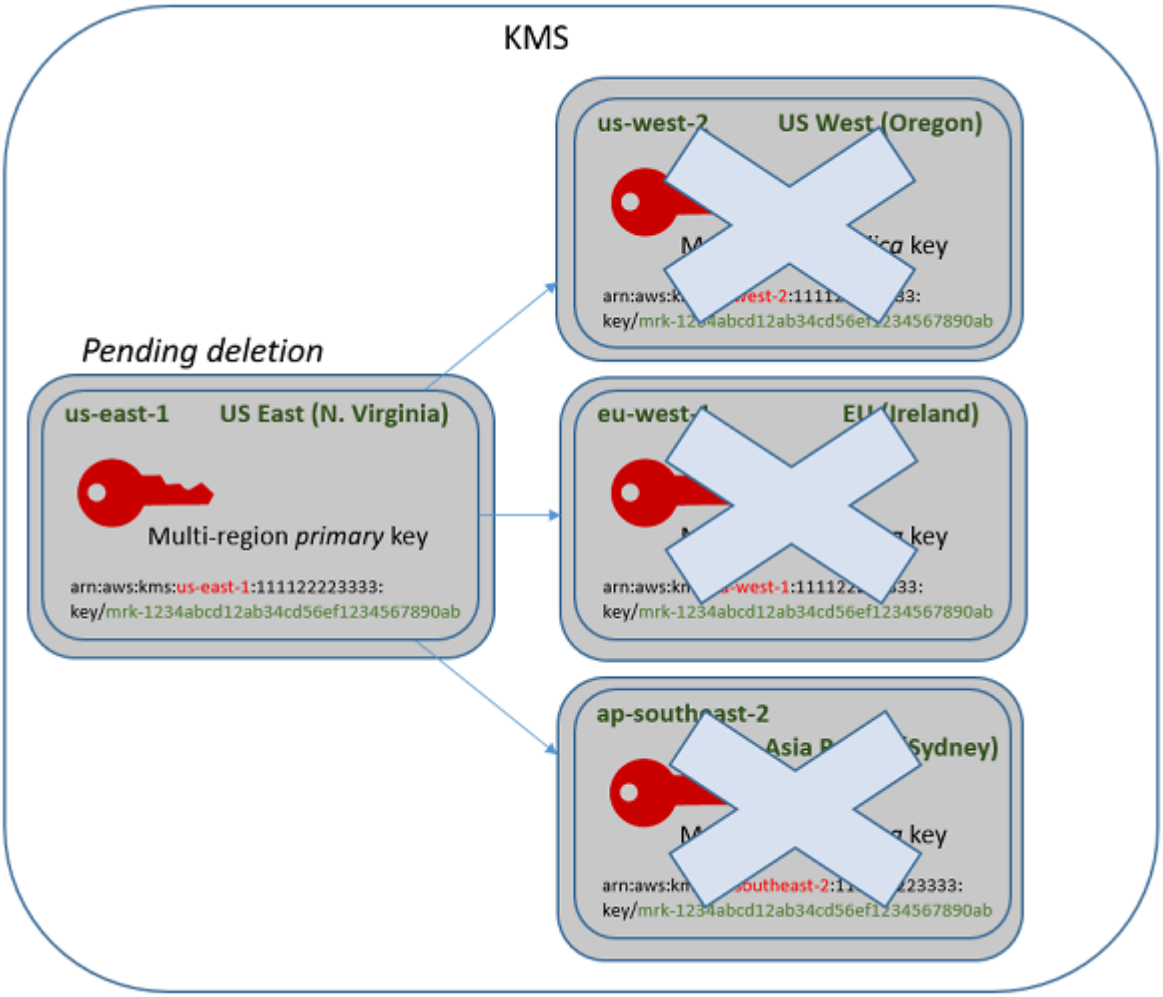
   Wenn Sie beispielsweise über drei Replikatschlüssel verfügen, können Sie das Löschen aller drei gleichzeitig planen. Sie können die gleichen oder unterschiedliche Wartezeiten haben. Beachten Sie, dass die Wartezeit für den Primärschlüssel noch nicht begonnen hat. Sein Schlüsselstatus ist `PendingReplicaDeletion`, da er über vorhandene Replikatschlüssel verfügt.

```
KMS key:                    Key state:
Primary key (us-east-1)       Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Pending deletion (7 days)
```

```
Replica (eu-west-1)           Pending deletion (7 days)
Replica (ap-southeast-2)      Pending deletion (30 days)
```

3. Sie können das geplante Löschen des Primärschlüssels oder eines Replikatschlüssels abbrechen, bis er gelöscht wird. Der Schlüsselstatus wechselt zu `Disabled`, aber Sie können den KMS-Schlüssel erneut aktivieren.
4. Wenn die Wartezeit des letzten Replikatschlüssels abgelaufen ist, löscht AWS KMS den letzten Replikatschlüssel. Der Schlüsselstatus des Primärschlüssels ändert sich von `Pending replica deletion` (`PendingReplicaDeletion`) zu `Pending deletion` (`PendingDeletion`) und die 7- bis 30-Tage-Wartezeit für den Primärschlüssel beginnt.

```
KMS key:                    Key state:
Primary key (us-east-1)       Pending deletion (waiting period 30 days)
```



5. Wenn die Wartezeit abgelaufen ist, löscht AWS KMS den Primärschlüssel.

Die Mindestwartezeit zum Löschen eines Primärschlüssels mit Replikaten beträgt 14 Tage.

Wenn Sie das Löschen des Primärschlüssels und aller Replikatschlüssel mit einer Wartezeit von 7 Tagen planen, werden die Replikatschlüssel nach 7 Tagen gelöscht. Der Primärschlüssel wird am 14. Tag gelöscht.

- Tag 1: Planen Sie das Löschen der Primär- und Replikatschlüssel mit der minimalen Wartezeit von 7 Tagen. Die 7-Tage-Wartezeiten für das Löschen der Replikatschlüssel werden gestartet. Die Wartezeit für den Primärschlüssel wird noch nicht gestartet.
- Tag 7: Die Wartezeiten für das Löschen der Replikatschlüssel enden. AWS KMS löscht alle Replikatschlüssel. Wenn der letzte Replikatschlüssel gelöscht wird, beginnt die 7-Tage-Wartezeit für den Primärschlüssel.
- Tag 14: Die Wartezeit für den Primärschlüssel endet. AWS KMS löscht den Primärschlüssel.

Sie können einen Datensatz Ihrer Aktionen in Ihrem AWS CloudTrail-Protokoll anzeigen. AWS KMS zeichnet die Produktionen auf, die das Löschen eines jeden KMS-Schlüssels planen und die Aktion, die den KMS-Schlüssel löscht.

Löschen eines Primärschlüssels (Konsole)

Gehen Sie wie folgt vor, um einen multiregionalen Primärschlüssel zu löschen.

So planen Sie die Löschung

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Markieren Sie das Kontrollkästchen neben dem Primärschlüssel, den Sie löschen möchten. Sie können auch einen oder mehrere KMS-Schlüssel auswählen, einschließlich der Replikate dieses Primärschlüssels.
5. Wählen Sie Key actions (Schlüsselaktionen), Schedule key deletion (Schlüssellöschung planen).
6. Lesen und berücksichtigen Sie die Warnung und die Informationen zum Abbrechen des Löschens während der Wartezeit. Wenn Sie den Löschvorgang abbrechen möchten, wählen Sie Cancel (Abbrechen).

7. Geben Sie für Waiting period (in days) (Wartezeit (in Tagen)) eine Anzahl von Tagen zwischen 7 und 30 ein. Wenn Sie mehrere KMS-Schlüssel ausgewählt haben, gilt die von Ihnen gewählte Wartezeit für alle ausgewählten KMS-Schlüssel. Die Wartezeit für Replikatschlüssel wird gleichzeitig ausgeführt, aber die Wartezeit für den Primärschlüssel beginnt erst wenn AWS KMS den letzten Replikatschlüssel löscht.
8. Aktivieren Sie zur Bestätigung, dass Sie den Schlüssel löschen wollen, das Kontrollkästchen neben Confirm that you want to delete this key in ***<number of days>***.
9. Wählen Sie Schedule deletion.

Um den Löschstatus Ihrer KMS-Schlüssel zu überprüfen, sehen Sie auf der Detailseite für den Primärschlüssel unter General configuration (allgemeine Konfiguration). Der Schlüsselstatus wird im Dialogfeld Status angezeigt. Wenn sich der Schlüsselstatus des Primärschlüssels in `Pending deletion` ändert, wird das geplante Löschdatum angezeigt.

Sie können auch den Schlüsselstatus (Status) aller Primär- und Replikatschlüssel auf der Registerkarte Regionality (Regionalität) der Detailseite eines beliebigen multiregionalen Schlüssels überprüfen. Details hierzu finden Sie unter Anzeigen von multiregionalen Schlüsseln.

Löschen eines Primärschlüssels (AWS KMS-API)

Um einen multiregionalen Replikatschlüssel zu löschen, verwenden Sie die -ScheduleKeyDeletionOperation. Um den KMS-Schlüssel anzugeben, verwenden Sie dessen Schlüssel-ID oder Schüssel-ARN. Wenn Sie mit multiregionalen Schlüsseln arbeiten, können Sie die Häufigkeit von Fehlern reduzieren, indem Sie den Schlüssel-ARN mit dem expliziten Regions-Wert verwenden.

Dieser Befehl löscht beispielsweise einen Primärschlüssel aus der Region us-east-1 (USA Ost (Nord-Virginia)). Da der Befehl keine Wartezeit angibt, wird die Wartezeit auf den Standardwert von 30 Tagen festgelegt.

```
$ aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt er den Schlüssel-ARN, den resultierenden Schlüsselstatus und die Wartezeit (`PendingWindowInDays`) zurück.

Wenn der Primärschlüssel keine Replikate hat, ist der Schlüsselstatus des Primärschlüssels `PendingDeletion` und die Ausgabe enthält das `DeletionDate`-Feld. Wenn Replikatschlüssel

verbleiben, ist der Schlüsselstatus des Primärschlüssels `PendingReplicaDeletion` und `DeletionDate` wird weggelassen, weil es unsicher ist. Selbst wenn die Replikatschlüssel ebenfalls zum Löschen geplant sind, können Sie den geplanten Löschvorgang abbrechen.

Stellen Sie beim Löschen eines multiregionalen Primärschlüssels sicher, dass die Schlüssel-ID und die Regions-Werte im Schlüssel-ARN die erwarteten Werte sind.

```
{
    "KeyId": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyState": "PendingReplicaDeletion",
    "PendingWindowInDays": 30
}
```

Um den Löschstatus Ihrer KMS-Schlüssel zu überprüfen, verwenden Sie die -DescribeKeyOperation für den Primärschlüssel oder alle verbleibenden Replikatschlüssel. Die Wartezeit für den Primärschlüssel wird erst gestartet, wenn das letzte Replikat gelöscht wird und der Schlüsselstatus in `PendingDeletion` ändert.

Um das erwartete Löschdatum des Primärschlüssels zu berechnen, durchlaufen Sie die Replikatschlüssel-ARNs in der Antwort, führen Sie `DescribeKey` auf jedem aus, erhalten Sie den neuesten `DeletionDate`-Wert und geben Sie dann den `PendingDeletionWindowInDays`-Wert für den Primärschlüssel hinzu. Die Wartezeiten für die Replikatschlüssel werden gleichzeitig ausgeführt.

Im folgenden Beispiel ist der KMS-Schlüssel ein multiregionaler Primärschlüssel mit vorhandenen Replikatschlüsseln. Da der Schlüsselstatus `PendingReplicaDeletion` ist, enthält die Antwort die Wartezeit (`PendingWindowInDays`), aber nicht das `DeletionDate`. Das tatsächliche Löschdatum des Primärschlüssels hängt davon ab, wann die Replikatschlüssel gelöscht werden.

```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1597902361.481,
```

```
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingReplicaDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "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": [
                {
                    "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "us-west-2"
                },
{
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                }
            ]
        },
        "PendingDeletionWindowInDays": 30
    }
}
```

Wenn alle Replikate gelöscht werden, zeigt die `DescribeKey`-Ausgabe den verbleibenden Primärschlüssel mit dem Schlüsselstatus `PendingDeletion` an. Während der Schlüsselstatus

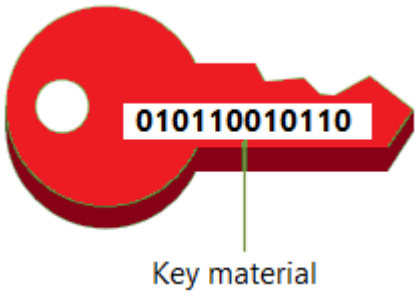
`PendingDeletion` ist, erscheint das `DeletionDate`-Feld anstelle des `PendingWindowInDays`-Felds.

```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "",
        "CreationDate": 1597902361.481,
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "DeletionDate": 1597968000.0,
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "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": []
        }
    }
}
```

Schlüsselmaterial für AWS KMS Schlüssel importieren

Sie können einen AWS KMS keys (KMS-Schlüssel) mit Schlüsselmaterial erstellen, das Sie bereitstellen.

Ein KMS-Schlüssel ist eine logische Darstellung eines Verschlüsselungsschlüssels. Die Metadaten für einen KMS-Schlüssel enthalten die ID des Schlüsselmaterials, das zur Ver- und Entschlüsselung von Daten verwendet wird. Beim Erstellen eines KMS-Schlüssels generiert AWS KMS standardmäßig das Schlüsselmaterial für diesen KMS-Schlüssel. Aber Sie können einen KMS-Schlüssel ohne Schlüsselmaterial erstellen und dann Ihr eigenes Schlüsselmaterial in diesen KMS-Schlüssel importieren, ein Feature, das oft als Bring Your Own Key (BYOK, bringe deinen eigenen Schlüssel) bezeichnet wird.



> **Note**
>
> AWS KMS unterstützt nicht die Entschlüsselung von AWS KMS Chiffretext außerhalb von AWS KMS, selbst wenn der Chiffretext unter einem KMS-Schlüssel mit importiertem Schlüsselmaterial verschlüsselt wurde. AWS KMS veröffentlicht das für diese Aufgabe erforderliche Chiffretextformat nicht, und das Format kann sich ohne vorherige Ankündigung ändern.

Importiertes Schlüsselmaterial wird für alle Arten von KMS-Schlüsseln unterstützt, mit Ausnahme von KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern.

Wenn Sie importiertes Schlüsselmaterial verwenden, bleiben Sie für das Schlüsselmaterial verantwortlich und erlauben gleichzeitig, eine Kopie davon AWS KMS zu verwenden. Sie können dies aus den folgenden Gründe tun:

- Um nachzuweisen, dass das Schlüsselmaterial mit einer zufälligen Quelle generiert wurde, die Ihre Anforderungen erfüllt.
- Um Schlüsselmaterial aus Ihrer eigenen Infrastruktur mit AWS Diensten AWS KMS zu verwenden und den Lebenszyklus des darin enthaltenen Schlüsselmaterials zu verwalten AWS.
- Zur Verwendung vorhandener, etablierter Schlüssel AWS KMS, z. B. Schlüssel für Codesignatur, PKI-Zertifikatsignierung und per Zertifikat fixierte Anwendungen

- Um eine Ablaufzeit für das Schlüsselmaterial festzulegen AWS und es manuell zu löschen, es aber auch in future wieder verfügbar zu machen. Im Gegensatz dazu erfordert die Planung der Schlüssellöschung eine Wartezeit von 7 bis 30 Tagen. Danach können Sie den gelöschten KMS-Schlüssel nicht wiederherstellen.
- Die Originalkopie des Schlüsselmaterials zu besitzen und es AWS für zusätzliche Haltbarkeit und Notfallwiederherstellung während des gesamten Lebenszyklus des Schlüsselmaterials außerhalb des Betriebs aufzubewahren.
- Bei asymmetrischen Schlüsseln und HMAC-Schlüsseln werden beim Import kompatible und interoperable Schlüssel erstellt, die innerhalb und außerhalb von funktionieren. AWS

Sie können die Verwendung und Verwaltung eines KMS-Schlüssels mit importiertem Schlüsselmaterial prüfen und überwachen. AWS KMS zeichnet ein Ereignis in Ihrem AWS CloudTrail Protokoll auf, wenn Sie den KMS-Schlüssel erstellen, den öffentlichen Schlüssel und das Importtoken herunterladen und das Schlüsselmaterial importieren. AWS KMS zeichnet auch ein Ereignis auf, wenn Sie importiertes Schlüsselmaterial manuell löschen oder wenn abgelaufenes Schlüsselmaterial AWS KMS gelöscht wird.

Informationen zu wichtigen Unterschieden zwischen KMS-Schlüsseln mit importiertem Schlüsselmaterial und solchen, deren Schlüsselmaterial von generiert wurde AWS KMS, finden Sie unterInformationen zu importiertem Schlüsselmaterial.

Unterstützte KMS-Schlüssel

AWS KMS unterstützt importiertes Schlüsselmaterial für die folgenden Typen von KMS-Schlüsseln. Es ist jedoch nicht möglich, Schlüsselmaterial in KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern zu importieren.

- KMS-Schlüssel zur symmetrischen Verschlüsselung
- Asymmetrische RSA-KMS-Schlüssel (für Verschlüsselung oder Signierung, aber nicht für beides)
- Asymmetrische elliptische Kurven (ECC) KMS-Schlüssel (nur Signieren)
- Asymmetrische SM2-KMS-Schlüssel — nur für Regionen Chinas (zur Verschlüsselung oder Signierung, aber nicht für beide)
- HMAC-KMS-Schlüssel
- Schlüssel für mehrere Regionen aller unterstützten Typen.

Regionen

Importiertes Schlüsselmaterial wird in allem unterstützt AWS-Regionen , AWS KMS was unterstützt wird.

In den Regionen Chinas unterscheiden sich die wichtigsten Materialanforderungen für KMS-Schlüssel mit symmetrischer Verschlüsselung von denen in anderen Regionen. Details hierzu finden Sie unter Schritt 3 für den Import von Schlüsselmaterial: Verschlüsselung des Schlüsselmaterials.

Themen

Planung des Imports von Schlüsselmaterial

Mit importiertem Schlüsselmaterial können Sie Ihre AWS Ressourcen mit von Ihnen generierten kryptografischen Schlüsseln schützen. Das Schlüsselmaterial, das Sie importieren, ist einem bestimmten KMS-Schlüssel zugeordnet. Sie können dasselbe Schlüsselmaterial erneut in denselben KMS-Schlüssel importieren, aber Sie können kein anderes Schlüsselmaterial in den KMS-Schlüssel importieren und Sie können einen KMS-Schlüssel, der für importiertes Schlüsselmaterial entworfen wurde, nicht in einen KMS-Schlüssel mit AWS KMS Schlüsselmaterial konvertieren.

Weitere Informationen:

- the section called “Wählen Sie eine Spezifikation für den öffentlichen Verpackungsschlüssel”
- the section called “Auswählen des Verpackungsalgorithmus”

Themen

Informationen zu importiertem Schlüsselmaterial

Bevor Sie sich für den Import von Schlüsselmaterial in entscheiden AWS KMS, sollten Sie sich mit den folgenden Eigenschaften von importiertem Schlüsselmaterial vertraut machen.

Sie generieren das Schlüsselmaterial

Sie sind verantwortlich für die Generierung des Schlüsselmaterials mit einer zufälligen Quelle, die Ihre Anforderungen erfüllt.

Sie können das Schlüsselmaterial löschen.

Sie können importiertes Schlüsselmaterial aus einem KMS-Schlüssel löschen, wodurch der KMS-Schlüssel sofort unbrauchbar wird. Beim Importieren von Schlüsselmaterial in einen KMS-Schlüssel können Sie außerdem bestimmen, ob der Schlüssel abläuft und die Ablaufzeit festlegen. Wenn die Ablaufzeit erreicht ist, AWS KMS wird das Schlüsselmaterial gelöscht. Ohne Schlüsselmaterial kann der KMS-Schlüssel nicht in kryptografischen Operationen verwendet werden. Um den Schlüssel wiederherzustellen, müssen Sie das gleiche Schlüsselmaterial erneut in den Schlüssel importieren.

Das Schlüsselmaterial kann nicht geändert werden

Beim Importieren von Schlüsselmaterial in einen KMS-Schlüssel wird der KMS-Schlüssel dauerhaft diesem Schlüsselmaterial zugeordnet. Sie können dasselbe Schlüsselmaterial erneut importieren, aber kein anderes Schlüsselmaterial in diesen KMS-Schlüssel importieren. Außerdem ist es nicht möglich, die automatische Schlüsseldrehung für einen KMS-Schlüssel mit importiertem Schlüsselmaterial zu aktivieren. Sie können einen KMS-Schlüssel mit importiertem Schlüsselmaterial jedoch manuell drehen.

Sie können die Herkunft des Schlüsselmaterials nicht ändern

KMS-Schlüssel, die für importiertes Schlüsselmaterial entwickelt wurden, haben einen Ursprungswert von`EXTERNAL`, der nicht geändert werden kann. Sie können einen KMS-Schlüssel für importiertes Schlüsselmaterial nicht konvertieren, um Schlüsselmaterial aus einer anderen Quelle zu verwenden, einschließlich AWS KMS. Ebenso können Sie einen KMS-Schlüssel mit Schlüsselmaterial nicht in einen AWS KMS KMS-Schlüssel konvertieren, der für importiertes Schlüsselmaterial entworfen wurde.

Sie können kein Schlüsselmaterial exportieren

Sie können kein Schlüsselmaterial exportieren, das Sie importiert haben. AWS KMS Sie können das importierte Schlüsselmaterial in keiner Form an Sie zurücksenden. Sie müssen eine Kopie Ihres importierten Schlüsselmaterials außerhalb von AWS, vorzugsweise in einem

Schlüsselmanager, wie z. B. einem Hardware-Sicherheitsmodul (HSM), aufbewahren, damit Sie das Schlüsselmaterial erneut importieren können, falls Sie es löschen oder wenn es abläuft.

Sie können multiregionale Schlüssel mit importiertem Schlüsselmaterial erstellen

Multi-Region mit importiertem Schlüsselmaterial haben die Eigenschaften von KMS-Schlüsseln mit importiertem Schlüsselmaterial und können zwischen AWS-Regionen interoperieren. Um einen multiregionalen Schlüssel mit importiertem Schlüsselmaterial zu erstellen, müssen Sie dasselbe Schlüsselmaterial in den KMS-Schlüssel und in jeden Replikatschlüssel importieren. Details hierzu finden Sie unter Schlüsselmaterial in multiregionale Schlüssel importieren.

Asymmetrische Schlüssel und HMAC-Schlüssel sind portabel und interoperabel

Sie können Ihr asymmetrisches Schlüsselmaterial und Ihr HMAC-Schlüsselmaterial außerhalb von verwenden, um mit AWS KMS Schlüsseln AWS zu interagieren, die dasselbe importierte Schlüsselmaterial enthalten.

Im Gegensatz zum AWS KMS symmetrischen Chiffretext, der untrennbar mit dem im Algorithmus verwendeten KMS-Schlüssel verbunden ist, werden standardmäßige HMAC-Formate und asymmetrische Formate für Verschlüsselung, AWS KMS Signierung und MAC-Generierung verwendet. Dadurch sind die Schlüssel übertragbar und unterstützen herkömmliche Escrow-Key-Szenarien.

Wenn Ihr KMS-Schlüssel Schlüsselmaterial importiert hat, können Sie das importierte Schlüsselmaterial außerhalb von verwenden, um die folgenden Operationen auszuführen. AWS

- HMAC-Schlüssel – Sie können ein HMAC-Tag, das durch den HMAC-KMS-Schlüssel generiert wurde, mit importiertem Schlüsselmaterial überprüfen. Sie können den HMAC-KMS-Schlüssel auch zusammen mit dem importierten Schlüsselmaterial verwenden, um ein HMAC-Tag zu verifizieren, das mit dem Schlüsselmaterial außerhalb von generiert wurde. AWS
- Asymmetrische Verschlüsselungsschlüssel — Sie können Ihren privaten asymmetrischen Verschlüsselungsschlüssel außerhalb von verwenden, AWS um einen durch den KMS-Schlüssel verschlüsselten Chiffretext mit dem entsprechenden öffentlichen Schlüssel zu entschlüsseln. Sie können Ihren asymmetrischen KMS-Schlüssel auch verwenden, um einen asymmetrischen Chiffretext zu entschlüsseln, der außerhalb von generiert wurde. AWS
- Asymmetrische Signaturschlüssel — Sie können Ihren asymmetrischen KMS-Schlüssel mit importiertem Schlüsselmaterial verwenden, um digitale Signaturen zu überprüfen, die mit Ihrem privaten Signaturschlüssel außerhalb von generiert wurden. AWS Sie können Ihren asymmetrischen öffentlichen Signaturschlüssel auch außerhalb von verwenden, um Signaturen AWS zu überprüfen, die mit Ihrem asymmetrischen KMS-Schlüssel generiert wurden.

Wenn Sie dasselbe Schlüsselmaterial in verschiedene KMS-Schlüssel desselben AWS-Region importieren, sind diese Schlüssel ebenfalls interoperabel. Um interoperable KMS-Schlüssel in verschiedenen Sprachen zu erstellen AWS-Regionen, erstellen Sie einen Schlüssel für mehrere Regionen mit importiertem Schlüsselmaterial.

Symmetrische Verschlüsselungsschlüssel sind nicht portabel oder interoperabel

Die daraus resultierenden symmetrischen Chiffretexte sind weder portabel noch AWS KMS interoperabel. AWS KMS veröffentlicht nicht das symmetrische Chiffretext-Format, das für die Portabilität erforderlich ist, und das Format kann sich ohne vorherige Ankündigung ändern.

- AWS KMS kann symmetrische Chiffretexte, die Sie außerhalb von verschlüsseln, nicht entschlüsseln AWS, selbst wenn Sie importiertes Schlüsselmaterial verwenden.
- AWS KMS unterstützt nicht die Entschlüsselung von AWS KMS symmetrischem Chiffretext außerhalb von AWS KMS, selbst wenn der Chiffretext unter einem KMS-Schlüssel mit importiertem Schlüsselmaterial verschlüsselt wurde.
- KMS-Schlüssel mit demselben importierten Schlüsselmaterial sind nicht interoperabel. Der symmetrische Chiffretext, der Chiffretext AWS KMS generiert, der für jeden KMS-Schlüssel spezifisch ist. Dieses Geheimtextformat garantiert, dass nur der KMS-Schlüssel, der die Daten verschlüsselt hat, diese entschlüsseln kann.

Außerdem können Sie keine AWS Tools wie die clientseitige Verschlüsselung AWS Encryption SDKoder die clientseitige Amazon S3 S3-Verschlüsselung verwenden, um symmetrische Chiffretexte zu entschlüsseln AWS KMS .

Daher können Sie Schlüssel mit importiertem Schlüsselmaterial nicht zur Unterstützung von Schlüsseltreuhandvereinbarungen verwenden, bei denen ein autorisierter Dritter mit eingeschränktem Zugriff auf Schlüsselmaterial bestimmte Chiffretexte außerhalb von entschlüsseln kann. AWS KMS Um die Treuhandfunktion für Schlüssel zu unterstützen, verwenden Sie die AWS Encryption SDK, um Ihre Nachricht mit einem Schlüssel zu verschlüsseln, der unabhängig von AWS KMS ist.

Sie sind verantwortlich für Verfügbarkeit und Langlebigkeit

AWS KMS wurde entwickelt, um importiertes Schlüsselmaterial hochverfügbar zu halten. Die Haltbarkeit von importiertem Schlüsselmaterial wird jedoch AWS KMS nicht auf dem gleichen Niveau gehalten wie das AWS KMS generierte Schlüsselmaterial. Details hierzu finden Sie unter Schützen von importiertem Schlüsselmaterial.

Schützen von importiertem Schlüsselmaterial

Das Schlüsselmaterial, das Sie importieren, ist während des Transports und im Ruhezustand geschützt. Vor dem Import des Schlüsselmaterials verschlüsseln (oder „umhüllen“) Sie das Schlüsselmaterial mit dem öffentlichen Schlüssel eines RSA-Schlüsselpaars, das in AWS KMS Hardware-Sicherheitsmodulen (HSMs) generiert wurde, die im Rahmen des FIPS 140-2 Cryptographic Module Validation Program validiert wurden. Sie können das Schlüsselmaterial direkt mit dem öffentlichen Schlüssel zur Verpackung verschlüsseln oder das Schlüsselmaterial mit einem symmetrischen AES-Schlüssel verschlüsseln und anschließend den symmetrischen AES-Schlüssel mit dem öffentlichen RSA-Schlüssel verschlüsseln.

AWS KMS Entschlüsselt das Schlüsselmaterial nach Erhalt mit dem entsprechenden privaten Schlüssel in einem AWS KMS HSM und verschlüsselt es erneut unter einem symmetrischen AES-Schlüssel, der nur im flüchtigen Speicher des HSM vorhanden ist. Ihr Schlüsselmaterial verlässt zu keiner Zeit Ihr HSM im Klartext. Es wird nur während der Verwendung und nur innerhalb von HSMs entschlüsselt. AWS KMS

Die Verwendung Ihres KMS-Schlüssels mit importiertem Schlüsselmaterial hängt ausschließlich von den Zugriffskontrollrichtlinien ab, die Sie für den KMS-Schlüssel festlegen. Darüber hinaus können Sie Aliase und Tags verwenden, um den Zugriff auf den KMS-Schlüssel zu identifizieren und zu kontrollieren. Sie können den Schlüssel aktivieren und deaktivieren, seine Eigenschaften anzeigen und bearbeiten und ihn mithilfe von Diensten wie AWS CloudTrailüberwachen.

Sie behalten jedoch die einzige ausfallsichere Kopie Ihres Schlüsselmaterials. Im Gegenzug für dieses zusätzliche Maß an Kontrolle sind Sie für die Haltbarkeit und allgemeine Verfügbarkeit des importierten Schlüsselmaterials verantwortlich. AWS KMS ist darauf ausgelegt, importiertes Schlüsselmaterial hochverfügbar zu halten. Die Haltbarkeit von importiertem Schlüsselmaterial wird jedoch AWS KMS nicht auf dem gleichen Niveau gehalten wie das AWS KMS generierte Schlüsselmaterial.

Dieser Unterschied in der Haltbarkeit ist in den folgenden Fällen von Bedeutung:

- Wenn Sie eine Ablaufzeit für Ihr importiertes Schlüsselmaterial festlegen, AWS KMS wird das Schlüsselmaterial nach Ablauf gelöscht. AWS KMS löscht den KMS-Schlüssel oder seine Metadaten nicht. Sie können einen CloudWatch Amazon-Alarm erstellen, der Sie benachrichtigt, wenn importiertes Schlüsselmaterial sich dem Ablaufdatum nähert.

Sie können kein Schlüsselmaterial löschen, das für einen KMS-Schlüssel AWS KMS generiert wurde, und Sie können nicht festlegen, dass AWS KMS Schlüsselmaterial abläuft, obwohl Sie es rotieren können.

- Wenn Sie importiertes Schlüsselmaterial manuell AWS KMS löschen, wird das Schlüsselmaterial gelöscht, der KMS-Schlüssel oder seine Metadaten werden jedoch nicht gelöscht. Im Gegensatz dazu erfordert die Planung der Schlüssellöschung eine Wartezeit von 7 bis 30 Tagen, nach der AWS KMS den KMS-Schlüssel, seine Metadaten und sein Schlüsselmaterial löscht.
- In dem unwahrscheinlichen Fall, dass bestimmte regionsweite Ausfälle auftreten, die Auswirkungen haben AWS KMS (z. B. ein vollständiger Stromausfall), AWS KMS kann Ihr importiertes Schlüsselmaterial nicht automatisch wiederhergestellt werden. Der KMS-Schlüssel und seine Metadaten AWS KMS können jedoch wiederhergestellt werden.

Sie müssen eine Kopie des importierten Schlüsselmaterials außerhalb eines von AWS Ihnen kontrollierten Systems aufbewahren. Wir empfehlen, eine exportierbare Kopie des importierten Schlüsselmaterials in einem Schlüsselverwaltungssystem, z. B. einem HSM, zu speichern. Wenn Ihr importiertes Schlüsselmaterial gelöscht wird oder abläuft, ist der zugehörige KMS-Schlüssel unbrauchbar, bis Sie dasselbe Schlüsselmaterial erneut importieren. Wenn Ihr importiertes Schlüsselmaterial dauerhaft verloren geht, ist jeder unter dem KMS-Schlüssel verschlüsselte Geheimtext nicht wiederherstellbar.

Berechtigungen zum Importieren von Schlüsselmaterial

Um KMS-Schlüssel mit importiertem Schlüsselmaterial zu erstellen und zu verwalten, benötigt der Benutzer die Berechtigung für die Operationen in diesem Prozess. Sie können die `kms:GetParametersForImport`-, `kms:ImportKeyMaterial`-, und`kms:DeleteImportedKeyMaterial`-Berechtigungen in der Schlüsselrichtlinie beim Erstellen des KMS-Schlüssels bereitstellen. In der AWS KMS Konsole werden diese Berechtigungen automatisch für Schlüsseladministratoren hinzugefügt, wenn Sie einen Schlüssel mit einem externen Schlüsselmaterial-Ursprung erstellen.

Um KMS-Schlüssel mit importiertem Schlüsselmaterial zu erstellen, benötigt der Prinzipal die folgenden Berechtigungen.

- kms: CreateKey (IAM-Richtlinie)
  - Um diese Berechtigung auf KMS-Schlüssel mit importiertem Schlüsselmaterial zu beschränken, verwenden Sie die KeyOrigin Richtlinienbedingung kms: mit dem Wert. `EXTERNAL`

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- kms: GetParametersForImport (Schlüsselrichtlinie oder IAM-Richtlinie)
  - Um diese Berechtigung auf Anfragen zu beschränken, die einen bestimmten Wrapping-Algorithmus und eine bestimmte Wrapping-Schlüsselspezifikation verwenden, verwenden Sie die WrappingKeySpec Richtlinienbedingungen kms: WrappingAlgorithm und kms:.
- kms: ImportKeyMaterial (Schlüsselrichtlinie oder IAM-Richtlinie)
  - Verwenden Sie die ValidTo Richtlinienbedingungen kms: und kms:, um Schlüsselmaterial, das abläuft, zuzulassen oder zu verbieten ExpirationModel und das Ablaufdatum zu kontrollieren.

Um importiertes Schlüsselmaterial erneut zu importieren, benötigt der Principal die Berechtigungen kms: GetParametersForImport und kms: ImportKeyMaterial.

Um importiertes Schlüsselmaterial zu löschen, benötigt der Principal die kms: DeleteImportedKeyMaterial -Berechtigung.

Um beispielsweise der Beispiel-`KMSAdminRole` die Berechtigung zu erteilen, alle Aspekte eines KMS-Schlüssels mit importiertem Schlüsselmaterial zu verwalten, fügen Sie eine Schlüsselrichtlinienanweisung wie die folgende in die Schlüsselrichtlinie des KMS-Schlüssels ein.

```
{
  "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"
  ]
}
```

Anforderungen an importiertes Schlüsselmaterial

Das Schlüsselmaterial, das Sie importieren, muss mit der Schlüsselspezifikation des zugehörigen KMS-Schlüssels kompatibel sein. Importieren Sie bei asymmetrischen Schlüsselpaaren nur den privaten Schlüssel des Paares. AWS KMS leitet den öffentlichen Schlüssel vom privaten Schlüssel ab.

AWS KMS unterstützt die folgenden Schlüsselspezifikationen für KMS-Schlüssel mit importiertem Schlüsselmaterial.

KMS-Schlüssel-Schlüsselspezifikation	Schlüsselmaterialanforderungen
Schlüssel zur symmetrischen Verschlüsselung SYMMETRIC_DEFAULT	256 Bit (32 Byte) an Binärdaten In chinesischen Regionen muss es sich um 128 Bit (16 Byte) an Binärdaten handeln.
HMAC-Schlüssel HMAC_224 HMAC_256 HMAC_384 HMAC_512	Das HMAC-Schlüsselmaterial muss RFC 2104 entsprechen. Die Schlüssellänge muss mit der in der Schlüsselspezifikation angegebenen Länge übereinstimmen.
Asymmetrischer privater RSA-Schlüssel RSA_2048 RSA_3072 RSA_4096	Der asymmetrische private RSA-Schlüssel, den Sie importieren, muss Teil eines Schlüsselpaares sein, das RFC 3447 entspricht. Modul: 2 048 Bit, 3 072 Bit oder 4 096 Bit Anzahl der Primzahlen: 2 (RSA-Schlüssel mit mehreren Primzahlen werden nicht unterstützt)

KMS-Schlüssel-Schlüsselspezifikation	Schlüsselmaterialanforderungen
	Asymmetrisches Schlüsselmaterial muss im Format Public-Key Cryptography Standards (PKCS) #8, das RFC 5208 entspricht, BER- oder DER-codiert sein.
Asymmetrischer privater Schlüssel mit elliptisc her Kurve ECC_NIST_P256 (secp256r1) ECC_NIST_P384 (secp384r1) ECC_NIST_P521 (secp521r1) ECC_SECG_P256K1 (secp256k1)	Der asymmetrische private ECC-Schlüssel, den Sie importieren, muss Teil eines Schlüsselpaares sein, das RFC 5915 entspricht. Kurve: NIST P-256, NIST P-384, NIST P-521, oder Secp256k1 Parameter: Nur benannte Kurven (ECC-Schlüssel mit expliziten Parametern werden zurückgewiesen) Öffentliche Punktkoordinaten: Können komprimiert, unkomprimiert oder projektiv sein Asymmetrisches Schlüsselmaterial muss im Format Public-Key Cryptography Standards (PKCS) #8, das RFC 5208 entspricht, BER- oder DER-codiert sein.

KMS-Schlüssel-Schlüsselspezifikation	Schlüsselmaterialanforderungen
Asymmetrischer privater SM2-Schlüssel (nur Regionen Chinas)	Der asymmetrische private SM2-Schlüssel, den Sie importieren, muss Teil eines key pair sein, das GM/T 0003 entspricht. Kurve: SM2 Parameter: Nur benannte Kurve (SM2-Schlüssel mit expliziten Parametern werden zurückgewiesen) Öffentliche Punktkoordinaten: Können komprimiert, unkomprimiert oder projektiv sein Asymmetrisches Schlüsselmaterial muss im Format Public-Key Cryptography Standards (PKCS) #8, das RFC 5208 entspricht, BER- oder DER-codiert sein.

Verwalten von importiertem Schlüsselmaterial

Diese Themen erklären, wie man Schlüsselmaterial in einen KMS-Schlüssel importiert und wieder importiert und wie man importiertes Schlüsselmaterial erstellt, das automatisch abläuft.

Themen

- Überblick über den Import von Schlüsselmaterial
- Reimportieren von Schlüsselmaterial
- Identifizieren von KMS-Schlüsseln mit importiertem Schlüsselmaterial
- Einen CloudWatch Alarm für den Ablauf von importiertem Schlüsselmaterial erstellen
- Löschen von importiertem Schlüsselmaterial
- Löschen eines KMS-Schlüssels mit importiertem Schlüsselmaterial

Überblick über den Import von Schlüsselmaterial

Die folgende Übersicht erläutert den Prozess zum Importieren Ihres Schlüsselmaterials in AWS KMS. Weitere Informationen zu jedem einzelnen Vorgang finden Sie unter den entsprechenden Themen.

1. Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial – Der Ursprung muss `EXTERNAL` sein. Ein Schlüsselursprung von `EXTERNAL` gibt an, dass der Schlüssel für importiertes Schlüsselmaterial konzipiert ist und verhindert, dass Schlüsselmaterial für den KMS-Schlüssel generiert wird. AWS KMS In einem späteren Schritt importieren Sie Ihr eigenes Schlüsselmaterial in diesen KMS-Schlüssel.

   Das Schlüsselmaterial, das Sie importieren, muss mit der Schlüsselspezifikation des zugehörigen AWS KMS Schlüssels kompatibel sein. Weitere Informationen zur Kompatibilität finden Sie unter the section called “Anforderungen an importiertes Schlüsselmaterial”.
2. Herunterladen des öffentlichen Verpackungsschlüssels und Import-Tokens – Nachdem Sie Schritt 1 abgeschlossen haben, laden Sie einen öffentlichen Verpackungsschlüssel und einen Import-Token herunter. Diese Artikel schützen Ihr Schlüsselmaterial, während es importiert wird. AWS KMS

   In diesem Schritt wählen Sie den Typ („Schlüsselspezifikation“) des RSA-Verpackungsschlüssels und den Verpackungsalgorithmus, den Sie zur Verschlüsselung Ihrer Daten während der Übertragung an AWS KMS verwenden werden. Sie können jedes Mal, wenn Sie dasselbe Schlüsselmaterial importieren oder erneut importieren, eine andere Verpackungsschlüsselspezifikation und einen anderen Verpackungsschlüsselalgorithmus wählen.
3. Verschlüsseln des Schlüsselmaterials – Verwenden Sie den öffentlichen Verpackungsschlüssel, den Sie in Schritt 2 heruntergeladen haben, um das Schlüsselmaterial zu verschlüsseln, das Sie in Ihrem eigenen System erstellt haben.
4. Importieren von Schlüsselmaterial – Laden Sie das verschlüsselte Schlüsselmaterial, das Sie in Schritt 3 erstellt haben, und das in Schritt 2 heruntergeladene Import-Token hoch.

   In diesem Stadium können Sie optional eine Ablaufzeit festlegen. Wenn importiertes Schlüsselmaterial abläuft, wird es AWS KMS gelöscht und der KMS-Schlüssel wird unbrauchbar. Wenn Sie den KMS-Schlüssel erneut nutzen möchten, müssen Sie das gleiche Schlüsselmaterial erneut importieren.

   Wenn der Importvorgang erfolgreich abgeschlossen wurde, ändert sich der Schlüsselstatus des KMS-Schlüssels von `PendingImport` zu `Enabled`. Sie können Ihren KMS-Schlüssel jetzt in kryptografischen Operationen verwenden.

AWS KMS zeichnet einen Eintrag in Ihrem AWS CloudTrail Protokoll auf, wenn Sie den KMS-Schlüssel erstellen, den öffentlichen Schlüssel und das Importtoken herunterladen und das Schlüsselmaterial importieren. AWS KMS zeichnet auch einen Eintrag auf, wenn Sie importiertes Schlüsselmaterial löschen oder wenn abgelaufenes Schlüsselmaterial AWS KMS gelöscht wird.

Reimportieren von Schlüsselmaterial

Wenn Sie einen KMS-Schlüssel mit importiertem Schlüsselmaterial verwalten, müssen Sie das Schlüsselmaterial möglicherweise erneut importieren. Sie können das Schlüsselmaterial erneut importieren, um ablaufendes odergelöschtes Schlüsselmaterial zu ersetzen oder um das Ablaufmodell oder das Ablaufdatum des Schlüsselmaterials zu ändern.

Beim Importieren von Schlüsselmaterial in einen KMS-Schlüssel wird der KMS-Schlüssel dauerhaft diesem Schlüsselmaterial zugeordnet. Sie können dasselbe Schlüsselmaterial erneut importieren, aber kein anderes Schlüsselmaterial in diesen KMS-Schlüssel importieren. Sie können das Schlüsselmaterial nicht drehen und AWS KMS kann kein Schlüsselmaterial für einen KMS-Schlüssel mit importiertem Schlüsselmaterial erstellen.

Sie können Schlüsselmaterial jederzeit nach einem beliebigen Zeitplan, der Ihren Sicherheitsanforderungen entspricht, wieder importieren. Sie müssen nicht warten, bis das Schlüsselmaterial seine Ablaufzeit erreicht oder fast erreicht hat.

Verwenden Sie zum erneuten Importieren von Schlüsselmaterial das gleiche Verfahren wie beim erstmaligen Importieren des Schlüsselmaterials, jedoch mit den folgenden Ausnahmen.

- Verwenden Sie einen vorhandenen KMS-Schlüssel, anstatt einen neuen KMS-Schlüssel zu erstellen. Sie können Schritt 1 des Importvorgangs überspringen.
- Wenn Sie Schlüsselmaterial erneut importieren, können Sie das Ablaufmodell und das Ablaufdatum ändern.

Jedes Mal, wenn Sie Schlüsselmaterial in einen KMS-Schlüssel importieren, müssen Sie einen neuen Verpackungsschlüssel und ein Import-Token für den KMS-Schlüssel herunterladen und verwenden. Das Verpackungsverfahren hat keine Auswirkungen auf den Inhalt des Schlüsselmaterials. Sie können also unterschiedliche öffentliche Verpackungsschlüssel und unterschiedliche Verpackungsalgorithmen verwenden, um das gleiche Schlüsselmaterial zu importieren.

Identifizieren von KMS-Schlüsseln mit importiertem Schlüsselmaterial

Beim Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial ist der Wert der `Origin`-Eigenschaft des KMS-Schlüssels `EXTERNAL` und kann nicht geändert werden. Im Gegensatz zum Schlüsselstatus, hängt der `Origin`-Wert nicht vom Vorhandensein oder Fehlen von Schlüsselmaterial ab.

Sie können den `EXTERNAL`-Ursprungswert zur Identifizierung von KMS-Schlüsseln für importiertes Schlüsselmaterial verwenden. Sie können den Ursprung des Schlüssels in der AWS KMS Konsole oder mithilfe der DescribeKeyOperation ermitteln. Sie können auch die Eigenschaften des Schlüsselmaterials anzeigen, z. B. ob und wann es abläuft, indem Sie die Konsole oder die APIs verwenden.

Identifizieren von KMS-Schlüsseln mit importiertem Schlüsselmaterial (Konsole)

1. Öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Verwenden Sie eine der folgenden Methoden, um die `Origin`-Eigenschaft Ihrer KMS-Schlüssel anzuzeigen.

   - Um eine Origin (Ursprung)-Spalte zu Ihrer KMS-Schlüsseltabelle hinzuzufügen, wählen Sie das Symbol Settings (Einstellungen) in der oberen rechten Ecke. Wählen Sie Origin (Ursprung) und Confirm (Bestätigen) aus. In der Ursprung-Spalte sind KMS-Schlüssel mit einem Externen (Schlüsselmaterial importieren)-Ursprung-Eigenschaftswert leicht erkennbar.
   - Um den Wert der `Origin`-Eigenschaft eines bestimmten KMS-Schlüssels zu finden, wählen Sie die Schlüssel-ID oder den Alias für den KMS-Schlüssel aus. Wählen Sie die Registerkarte Cryptographic configuration (kryptografische Konfiguration) aus. Die Registerkarte wird unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.
4. Um detaillierte Informationen zum Schlüsselmaterial anzuzeigen, wählen Sie die Registerkarte Key material (Schlüsselmaterial) aus. Diese Registerkarte wird auf der Detailseite nur für KMS-Schlüssel mit importiertem Schlüsselmaterial angezeigt.

Um KMS-Schlüssel mit importiertem Schlüsselmaterial (API) zu identifizieren AWS KMS

Verwenden Sie die DescribeKeyOperation. Die Antwort enthält die `Origin`-Eigenschaft des KMS-Schlüssels, das Ablaufmodell und das Ablaufdatum, wie im folgenden Beispiel gezeigt.

```
$  aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Origin": "EXTERNAL",
        "ExpirationModel": "KEY_MATERIAL_EXPIRES"
        "ValidTo": 2023-06-05T12:00:00+00:00,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": 2018-06-09T00:06:50.831000+00:00,
        "Enabled": false,
        "MultiRegion": false,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Einen CloudWatch Alarm für den Ablauf von importiertem Schlüsselmaterial erstellen

Sie können einen CloudWatch Alarm erstellen, der Sie benachrichtigt, wenn sich das in einem KMS-Schlüssel importierte Schlüsselmaterial seiner Ablaufzeit nähert. Der Alarm kann Sie beispielsweise benachrichtigen, wenn die Zeit in weniger als 30 Tagen abläuft.

Wenn Sie Schlüsselmaterial in einen KMS-Schlüssel importieren, können Sie optional festlegen, an welchem Datum und zu welcher Uhrzeit das Schlüsselmaterial abläuft. Wenn das Schlüsselmaterial abläuft, wird das Schlüsselmaterial AWS KMS gelöscht und der KMS-Schlüssel wird unbrauchbar. Wenn Sie den KMS-Schlüssel erneut nutzen möchten, müssen Sie das Schlüsselmaterial erneut importieren. Wenn Sie das Schlüsselmaterial erneut importieren, bevor es abläuft, können Sie jedoch verhindern, dass Prozesse unterbrochen werden, die diesen KMS-Schlüssel verwenden.

Dieser Alarm verwendet die `SecondsUntilKeyMaterialExpires`Metrik, die AWS KMS veröffentlicht wird, um CloudWatch KMS-Schlüssel mit importiertem Schlüsselmaterial zu veröffentlichen, das abläuft. Jeder Alarm verwendet diese Metrik, um das importierte

Schlüsselmaterial für einen bestimmten KMS-Schlüssel zu überwachen. Es ist nicht möglich, einen einzelnen Alarm für alle KMS-Schlüssel mit ablaufendem Schlüsselmaterial oder einen Alarm für KMS-Schlüssel, die Sie möglicherweise in Zukunft erstellen werden, zu erstellen.

Voraussetzungen

Die folgenden Ressourcen sind für einen CloudWatch Alarm erforderlich, der den Ablauf von importiertem Schlüsselmaterial überwacht.

- Ein KMS-Schlüssel mit importiertem Schlüsselmaterial, das abläuft. Weitere Informationen dazu finden Sie unter Identifizieren von KMS-Schlüsseln mit importiertem Schlüsselmaterial.
- Amazon SNS-Thema. Einzelheiten finden Sie unter Erstellen eines Amazon SNS SNS-Themas im CloudWatch Amazon-Benutzerhandbuch.

Den Alarm erstellen

Folgen Sie den Anweisungen unter Erstellen Sie einen CloudWatch Alarm auf der Grundlage eines statischen Schwellenwerts und verwenden Sie dabei die folgenden erforderlichen Werte. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrik auswählen	Wählen Sie KMS und dann Per-Key Metrics (Metriken pro Schlüssel) aus. Wählen Sie die Zeile mit dem KMS-Schlüssel und der `SecondsUntilKeyMaterialExpires` -Metrik aus. Wählen Sie dann Select Metric (Metrik auswählen) aus. Die Liste Metrics (Metriken) zeigt die `SecondsUntilKeyMaterialExpires` -Metrik nur für KMS-Schlüssel an, deren importiertes Schlüsselmaterial abläuft. Wenn Sie keine KMS-Schlüssel mit diesen Eigenschaften im Konto und in der Region haben, ist diese Liste leer.
Statistik	Minimum
Intervall	1 Minute
Threshold-Typ	Statisch

Feld	Value (Wert)
Immer, wenn ...	Immer, wenn *metric-name* größer als 1 ist

Löschen von importiertem Schlüsselmaterial

Sie können das importierte Schlüsselmaterial jederzeit aus einem KMS-Schlüssel löschen. Außerdem wird das Schlüsselmaterial AWS KMS gelöscht, wenn importiertes Schlüsselmaterial mit einem Ablaufdatum abläuft. In beiden Fällen ändert sich der Schlüsselstatus des KMS-Schlüssels nach dem Löschen des Schlüsselmaterials in Import ausstehend und der KMS-Schlüssel kann erst dann für kryptografische Operationen verwendet werden, wenn Sie das gleiche Schlüsselmaterial erneut importieren. (Sie können kein anderes Schlüsselmaterial in den KMS-Schlüssel importieren).

Neben dem Deaktivieren des KMS-Schlüssels und dem Entziehen von Berechtigungen kann das Löschen von Schlüsselmaterial als Strategie verwendet werden, um die Verwendung des KMS-Schlüssels schnell, aber vorübergehend einzustellen. Im Gegensatz dazu wird bei der Planung des Löschens eines KMS-Schlüssels mit importiertem Schlüsselmaterial auch schnell die Verwendung des KMS-Schlüssels gestoppt. Wenn das Löschen jedoch während der Wartezeit nicht abgebrochen wird, werden der KMS-Schlüssel, das Schlüsselmaterial und alle wichtigen Metadaten dauerhaft gelöscht. Details hierzu finden Sie unter the section called “Löschen eines KMS-Schlüssels mit importiertem Schlüsselmaterial”.

Um Schlüsselmaterial zu löschen, können Sie die AWS KMS Konsole oder den DeleteImportedKeyMaterialAPI-Vorgang verwenden. AWS KMS zeichnet einen Eintrag in Ihrem AWS CloudTrail Protokoll auf, wenn Sie importiertes Schlüsselmaterial löschen und wenn abgelaufenes Schlüsselmaterial AWS KMS gelöscht wird.

Themen

- Wie sich das Löschen von Schlüsselmaterial auf Dienste auswirkt AWS
- Löschen von Schlüsselmaterial (Konsole)
- Schlüsselmaterial löschen (API)AWS KMS

Wie sich das Löschen von Schlüsselmaterial auf Dienste auswirkt AWS

Wenn Sie Schlüsselmaterial löschen, wird der KMS-Schlüssel ohne Schlüsselmaterial sofort unbrauchbar (je nach letztendlicher Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel

geschützten Datenschlüsseln verschlüsselt wurden, sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, dass viele von ihnen Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Löschen von Schlüsselmaterial (Konsole)

Sie können den verwenden AWS Management Console , um Schlüsselmaterial zu löschen.

1. Melden Sie sich bei der AWS Key Management Service (AWS KMS) -Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Führen Sie eine der folgenden Aktionen aus:
   - Aktivieren Sie das Kontrollkästchen für einen KMS-Schlüssel mit importiertem Schlüsselmaterial. Wählen Sie unter Key actions die Option Delete key material.
   - Wählen Sie den Alias oder die Schlüssel-ID des KMS-Schlüssels mit importiertem Schlüsselmaterial aus. Wählen Sie die Registerkarte Key material (Schlüsselmaterial) und dann Delete key material (Schlüsselmaterial löschen).
5. Bestätigen Sie, dass Sie das Schlüsselmaterial löschen möchten, und klicken Sie dann auf Delete key material. Der Zustand des KMS-Schlüssels, der dem Schlüsselstatus entspricht, ändert sich in Pending import (Import ausstehend).

Schlüsselmaterial löschen (API)AWS KMS

Um die AWS KMS API zum Löschen von Schlüsselmaterial zu verwenden, senden Sie eine DeleteImportedKeyMaterialAnfrage. Das folgende Beispiel zeigt, wie Sie dies mit dem AWS CLI tun können.

Ersetzen Sie *1234abcd-12ab-34cd-56ef-1234567890ab* mit der Schlüssel-ID des KMS-Schlüssels, dessen Schlüsselmaterial Sie löschen möchten. Für diese Operation können Sie die Schlüssel-ID oder den ARN des KMS-Schlüssels verwenden, aber keinen Alias.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Löschen eines KMS-Schlüssels mit importiertem Schlüsselmaterial

Das Löschen des Schlüsselmaterials eines KMS-Schlüssels mit importiertem Schlüsselmaterial ist temporär und reversibel. Um den Schlüssel wiederherzustellen, importieren Sie das zugehörige Schlüsselmaterial erneut.

Im Gegensatz dazu kann das Löschen eines KMS-Schlüssels nicht rückgängig gemacht werden. Wenn Sie das Löschen von Schlüsseln planen und die erforderliche Wartezeit abläuft, werden der KMS-Schlüssel, sein Schlüsselmaterial und alle mit dem KMS-Schlüssel verknüpften Metadaten AWS KMS dauerhaft und unwiderruflich gelöscht.

Das Risiko und die Folgen des Löschens eines KMS-Schlüssels mit importiertem Schlüsselmaterial hängen jedoch vom Typ („Schlüsselspezifikation“) des KMS-Schlüssels ab.

- Symmetrische Verschlüsselungsschlüssel – Wenn Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung löschen, können alle verbleibenden Geheimtexte, die mit diesem Schlüssel verschlüsselt wurden, nicht wiederhergestellt werden. Sie können keinen neuen symmetrischen KMS-Schlüssel erstellen, der die Geheimtexte eines gelöschten symmetrischen KMS-Schlüssels entschlüsseln kann, selbst wenn Sie dasselbe Schlüsselmaterial haben. Metadaten, die für jeden KMS-Schlüssel einzigartig sind, werden kryptografisch an jeden symmetrischen Geheimtext gebunden. Dieses Sicherheits-Feature garantiert, dass nur der KMS-Schlüssel, mit dem der symmetrische Geheimtext verschlüsselt wurde, ihn entschlüsseln kann, verhindert jedoch, dass Sie einen entsprechenden KMS-Schlüssel neu erstellen können.
- Asymmetrische Schlüssel und HMAC-Schlüssel — Wenn Sie über das ursprüngliche Schlüsselmaterial verfügen, können Sie einen neuen KMS-Schlüssel mit denselben kryptografischen Eigenschaften wie ein asymmetrischer oder gelöschter HMAC-KMS-Schlüssel erstellen. AWS KMS generiert standardmäßige RSA-Verschlüsselungstexte und -Signaturen, ECC-Signaturen und HMAC-Tags, die keine eindeutigen Sicherheitsmerkmale enthalten. Sie können auch einen HMAC-Schlüssel oder den privaten Schlüssel eines asymmetrischen Schlüsselpaares außerhalb von AWS verwenden.

  Ein neuer KMS-Schlüssel, den Sie mit demselben asymmetrischen oder HMAC-Schlüsselmaterial erstellen, hat eine andere Schlüssel-ID. Sie müssen eine neue Schlüsselrichtlinie erstellen, alle Aliase neu erstellen und bestehende IAM-Richtlinien und -Berechtigungen aktualisieren, damit sie auf den neuen Schlüssel verweisen.

Importieren von Schlüsselmaterial Schritt 1: Erstellen eines AWS KMS key ohne Schlüsselmaterial

AWS KMS erstellt standardmäßig Schlüsselmaterial für Sie, wenn Sie einen KMS-Schlüssel erstellen. Wenn Sie Ihr eigenes Schlüsselmaterial importieren möchten, können Sie beginnen, indem Sie zunächst einen KMS-Schlüssel ohne Schlüsselmaterial erstellen. Importieren Sie dann das Schlüsselmaterial. Um einen KMS-Schlüssel ohne Schlüsselmaterial zu erstellen, verwenden Sie die -AWS KMSKonsole oder die -CreateKeyOperation.

Um einen Schlüssel ohne Schlüsselmaterial zu erstellen, geben Sie einen Ursprung von `EXTERNAL` an. Die Herkunftseigenschaft eines KMS-Schlüssels ist unveränderlich. Sobald Sie einen Schlüssel erstellt haben, können Sie einen KMS-Schlüssel für importiertes Schlüsselmaterial nicht in einen KMS-Schlüssel mit Schlüsselmaterial aus AWS KMS oder irgendeiner anderen Quelle konvertieren.

Der Schlüsselstatus eines KMS-Schlüssels mit einem `EXTERNAL`-Ursprung und ohne Schlüsselmaterial ist `PendingImport`. Ein KMS-Schlüssel kann auf unbestimmte Zeit im Status `PendingImport`bleiben. Sie können jedoch einen KMS-Schlüssel im `PendingImport`-Status nicht für kryptografische Operationen verwenden. Wenn Sie Schlüsselmaterial importieren, ändert sich der Schlüsselstatus des KMS-Schlüssels zu `Enabled`, und Sie können ihn für kryptografische Operationen verwenden.

AWS KMS zeichnet ein Ereignis in Ihrem AWS CloudTrail Protokoll auf, wenn Sie den KMS-Schlüssel erstellen, den öffentlichen Schlüssel und das Import-Token herunterladen und das Schlüsselmaterial importieren. zeichnet AWS KMS auch ein CloudTrail Ereignis auf, wenn Sie importiertes Schlüsselmaterial löschen oder wenn abgelaufenes Schlüsselmaterial AWS KMSlöscht. ???

Weitere Informationen zum Erstellen von multiregionalen Schlüsseln mit importiertem Schlüsselmaterial finden Sie unter Schlüsselmaterial in multiregionale Schlüssel importieren.

Themen

- Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial (Konsole)
- Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial (AWS KMS-API)

Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial (Konsole)

Sie müssen nur einmal einen KMS-Schlüssel für das importierte Schlüsselmaterial anlegen. Sie können dasselbe Schlüsselmaterial so oft wie nötig in den bestehenden KMS-Schlüssel

importieren und wieder importieren, aber Sie können kein anderes Schlüsselmaterial in einen KMS-Schlüssel importieren. Details hierzu finden Sie unter Schritt 2: Herunterladen des öffentlichen Verpackungsschlüssels und des Import-Tokens.

Um vorhandene KMS-Schlüssel mit importiertem Schlüsselmaterial in Ihrer Tabelle mit den vom Kunden verwalteten Schlüsseln zu finden, verwenden Sie das Zahnradsymbol in der oberen rechten Ecke, um die Spalte Origin (Herkunft) in der Liste der KMS-Schlüssel anzuzeigen. Importierte Schlüssel haben den Herkunft-Wert Extern (Schlüsselmaterial importieren) erkennbar.

Um einen KMS-Schlüssel mit importiertem Schlüsselmaterial zu erstellen, folgen Sie zunächst den grundlegenden Anweisungen zum Erstellen eines KMS-Schlüssels Ihres bevorzugten Schlüsseltyps, mit der folgenden Ausnahme.

Nachdem Sie die Schlüsselnutzung ausgewählt haben, gehen Sie wie folgt vor:

1. Erweitern Sie Advanced options (Erweiterte Optionen).
2. Stellen Sie die Key material origin (Herkunft des Schlüsselmaterials) auf External (Import key material) (Extern (Schlüsselmaterial importieren)) ein.
3. Aktivieren Sie dann das Kontrollkästchen neben Mir ist bekannt, dass die Verwendung eines importieren Schlüssel sich auf die Sicherheit und Lebensdauer auswirken kann um zu bestätigen, dass Sie die Auswirkungen der Verwendung von importiertem Schlüsselmaterial kennen. Weitere Informationen zu diesen Implikationen finden Sie unter Schützen von importiertem Schlüsselmaterial.
4. Kehren Sie zu den grundlegenden Anweisungen zurück. Die verbleibenden Schritte des grundlegenden Verfahrens sind für alle KMS-Schlüssel dieses Typs identisch.

Wenn Sie Fertig stellen wählen, haben Sie einen KMS-Schlüssel ohne Schlüsselmaterial und mit dem Status (Schlüsselstatus) Ausstehender Import erstellt.

Anstatt jedoch zur Tabelle mit vom Kunden verwalteten Schlüsseln zurückzukehren, zeigt die Konsole eine Seite an, auf der Sie den öffentlichen Schlüssel herunterladen und das Token importieren können, das Sie für den Import Ihres Schlüsselmaterials benötigen. Sie können jetzt mit dem Download-Schritt fortfahren oder auf Abbrechen klicken, um den Vorgang an dieser Stelle zu beenden. Sie können jederzeit zu diesem Download-Schritt zurückkehren.

Danach: Schritt 2: Herunterladen des öffentlichen Verpackungsschlüssels und des Import-Tokens.

Erstellen eines KMS-Schlüssels ohne Schlüsselmaterial (AWS KMS-API)

Um die AWS KMS-API zum Erstellen eines KMS-Schlüssels mit symmetrischer Verschlüsselung ohne Schlüsselmaterial zu verwenden, senden Sie eine -CreateKeyAnforderung, bei der der Parameter auf `Origin` gesetzt ist`EXTERNAL`. Im folgenden Beispiel wird gezeigt, wie Sie dies mit dem AWS Command Line Interface (AWS CLI) durchführen.

```
$ aws kms create-key --origin EXTERNAL
```

Wenn der Befehl erfolgreich ausgeführt wurde, wird eine Ausgabe ähnlich der Folgenden angezeigt. Der `Origin` des AWS KMS-Schlüssels ist `EXTERNAL` und seine `KeyState` ist `PendingImport`.

> **Tip**
>
> Wenn der Befehl nicht erfolgreich ist, wird möglicherweise eine `KMSInvalidStateException` oder eine `NotFoundException` angezeigt. Sie können die Anfrage wiederholen.

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

Kopieren Sie den Wert `KeyId` für die spätere Verwendung aus der Befehlsausgabe und fahren Sie dann mit Schritt 2: Herunterladen des öffentlichen Verpackungsschlüssels und des Import-Tokens fort.

> **Note**
>
> Dieser Befehl erstellt einen KMS-Schlüssel für die symmetrische Verschlüsselung mit `KeySpec` von `SYMMETRIC_DEFAULT` und `KeyUsage` von `ENCRYPT_DECRYPT`. Sie können die optionalen Parameter `--key-spec` und `--key-usage` verwenden, um einen asymmetrischen oder HMAC-KMS-Schlüssel zu erstellen. Weitere Informationen finden Sie unter dem Vorgang CreateKey.

Schritt 2 für den Import von Schlüsselmaterial: Herunterladen des öffentlichen Verpackungsschlüssels und des Import-Tokens

Nachdem Sie ein AWS KMS key ohne Schlüsselmaterial erstellt haben, laden Sie mithilfe der AWS KMS Konsole oder der GetParametersForImportAPI einen öffentlichen Schlüssel zur Verpackung und ein Import-Token für diesen KMS-Schlüssel herunter. Der öffentliche Verpackungsschlüssel und das Import-Token bilden einen untrennbaren Satz, der zusammen verwendet werden muss.

Sie werden den öffentlichen Verpackungsschlüssel verwenden, um Ihr Schlüsselmaterial für den Transport zu verschlüsseln. Bevor Sie ein RSA-Wrapping-Schlüsselpaar herunterladen, wählen Sie die Länge (Schlüsselspezifikation) des RSA-Wrapping-Schlüsselpaars und den Wrapping-Algorithmus aus, mit dem Sie Ihr importiertes Schlüsselmaterial für den Transport in Schritt 3 verschlüsseln werden. AWS KMS unterstützt auch die SM2-Wrapping-Schlüsselspezifikation (nur für chinesische Regionen).

Jeder öffentliche Verpackungsschlüssel und Import-Token-Satz ist 24 Stunden gültig. Wenn Sie sie nicht innerhalb von 24 Stunden nach dem Download verwenden, um Schlüsselmaterial zu importieren, müssen Sie einen neuen Satz herunterladen. Sie können jederzeit neue öffentliche Verpackungsschlüssel herunterladen und Tokensätze importieren. Auf diese Weise können Sie die Länge Ihres RSA-Verpackungsschlüssels („Schlüsselspezifikation“) ändern oder einen verlorenen Satz ersetzen.

Sie können auch einen öffentlichen Verpackungsschlüssel und ein Import-Token-Set herunterladen, um dasselbe Schlüsselmaterial erneut in einen KMS-Schlüssel zu importieren. Sie können dies tun, um die Ablaufzeit für das Schlüsselmaterial festzulegen oder zu ändern, oder um abgelaufenes oder

gelöschtes Schlüsselmaterial wiederherzustellen. Sie müssen Ihr Schlüsselmaterial bei jedem Import in herunterladen und erneut verschlüsseln. AWS KMS

Verwendung des öffentlichen Verpackungsschlüssels

Der Download beinhaltet einen öffentlichen Schlüssel, der nur für Sie bestimmt ist. AWS-Konto Er wird auch als öffentlicher Schlüssel bezeichnet.

Bevor Sie Schlüsselmaterial importieren, verschlüsseln Sie das Schlüsselmaterial mit dem öffentlichen Wrapping-Schlüssel und laden das verschlüsselte Schlüsselmaterial anschließend in hoch. AWS KMS Beim AWS KMS Empfang Ihres verschlüsselten Schlüsselmaterials entschlüsselt es das Schlüsselmaterial mit dem entsprechenden privaten Schlüssel und verschlüsselt das Schlüsselmaterial anschließend erneut unter einem symmetrischen AES-Schlüssel, alles innerhalb eines AWS KMS Hardware-Sicherheitsmoduls (HSM).

Nutzen des Import-Tokens

Der Download enthält ein Import-Token mit Metadaten, das sicherstellt, dass Ihr Schlüsselmaterial korrekt importiert wird. Wenn Sie Ihr verschlüsseltes Schlüsselmaterial hochladen AWS KMS, müssen Sie dasselbe Import-Token hochladen, das Sie in diesem Schritt heruntergeladen haben.

Wählen Sie eine Spezifikation für den öffentlichen Verpackungsschlüssel

Um Ihr Schlüsselmaterial beim Import zu schützen, verschlüsseln Sie es mit einem öffentlichen Wrapping-Schlüssel, von dem Sie herunterladen AWS KMS, und einem unterstützten Wrapping-Algorithmus. Sie wählen eine Schlüsselspezifikation aus, bevor Sie Ihren öffentlichen Verpackungsschlüssel und das Import-Token herunterladen. Alle Wrapping-Schlüsselpaare werden in AWS KMS Hardware-Sicherheitsmodulen (HSM) generiert. Der private Schlüssel verlässt das HSM niemals im Klartext.

Die wichtigsten Spezifikationen für das RSA-Wrapping

Die Schlüsselspezifikation des öffentlichen Verpackungsschlüssels bestimmt die Länge der Schlüssel in dem RSA-Schlüsselpaar, das Ihr Schlüsselmaterial während des Transports zu AWS KMS schützt. Im Allgemeinen empfehlen wir die Verwendung des am längsten umschließenden öffentlichen Verpackungsschlüssels, der praktisch ist. Wir bieten verschiedene Spezifikationen für öffentliche Verpackungsschlüssel an, um eine Vielzahl von HSMs und Schlüsselmanagern zu unterstützen.

AWS KMS unterstützt die folgenden Schlüsselspezifikationen für die RSA-Wrapping-Schlüssel, die für den Import von Schlüsselmaterial aller Art verwendet werden, sofern nicht anders angegeben.

- RSA_4096 (bevorzugt)
- RSA_3072
- RSA_2048

> **Note**
>
> Die folgende Kombination wird NICHT unterstützt: ECC_NIST_P521-Schlüsselmaterial, die RSA_2048-Spezifikation für öffentliche Verpackungsschlüssel und ein RSAES_OAEP_SHA_*-Verpackungsalgorithmus.
> Sie können das Schlüsselmaterial von ECC_NIST_P521 nicht direkt mit einem öffentlichen RSA_2048-Verpackungsschlüssel verpacken. Verwenden Sie einen größeren Verpackungsschlüssel oder einen RSA_AES_KEY_WRAP_SHA_*-WRAP_*-Verpackungsalgorithmus.

Spezifikation für SM2-Wrapping-Schlüssel (nur für chinesische Regionen)

AWS KMS unterstützt die folgenden Schlüsselspezifikationen für SM2-Wickelschlüssel, die zum Import von asymmetrischem Schlüsselmaterial verwendet werden.

- SM2

Auswählen des Verpackungsalgorithmus

Um Ihr Schlüsselmaterial während des Imports zu schützen, verschlüsseln Sie es mit dem heruntergeladenen öffentlichen Verpackungsschlüssel und einem unterstützten Verpackungsalgorithmus.

AWS KMS unterstützt mehrere Standard-RSA-Wrapping-Algorithmen und einen zweistufigen Hybrid-Wrapping-Algorithmus. Im Allgemeinen empfehlen wir, den sichersten Verpackungsalgorithmus zu verwenden, der mit Ihrem importierten Schlüsselmaterial und den Verpackungsschlüsselspezifikationen kompatibel ist. In der Regel wählen Sie einen Algorithmus aus, der von dem Hardwaresicherheitsmodul (HSM) oder dem Schlüsselverwaltungssystem unterstützt wird, mit dem Ihr Schlüsselmaterial geschützt werden.

In der folgenden Tabelle sind die Verpackungsalgorithmen aufgeführt, die für jeden Typ von Schlüsselmaterial und KMS-Schlüssel unterstützt werden. Die Algorithmen werden in Präferenzreihenfolge aufgeführt.

Schlüsselmaterial	Unterstützter Verpackungsalgorithmus und -Spezifikation
Schlüssel zur symmetrischen Verschlüsselung 256-Bit-AES-Schlüssel 128-Bit-SM4-Schlüssel (nur chinesische Regionen)	Verpackungsalgorithmen: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Veraltete Verpackungsalgorithmen: RSAES_PKCS1_V1 **Note** Unterstützt seit dem 10. Oktober 2023 den Wrapping-Algorithmus RSAES_PKCS1_V1_5 AWS KMS nicht. Verpackungsschlüsselspezifikationen: RSA_2048 RSA_3072 RSA_4096
Asymmetrischer privater RSA-Schlüssel	Verpackungsalgorithmen: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE (nur Regionen Chinas)

Schlüsselmaterial	Unterstützter Verpackungsalgorithmus und -Spezifikation
	Verpackungsschlüsselspezifikationen: RSA_2048 RSA_3072 RSA_4096 SM2 (nur China-Regionen)
Asymmetrischer privater Schlüssel mit elliptischer Kurve (ECC) Sie können die RSAES_OAEP_SHA_*-Verpackungsalgorithmen nicht mit der RSA_2048-Verpackungsschlüsselspezifikation verwenden, um ECC_NIST_P521-Schlüsselmaterial zu verpacken.	Verpackungsalgorithmen: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (nur Regionen in China) Verpackungsschlüsselspezifikationen: RSA_2048 RSA_3072 RSA_4096 SM2 (nur China-Regionen)

Schlüsselmaterial	Unterstützter Verpackungsalgorithmus und -Spezifikation
Asymmetrischer privater SM2-Schlüssel (nur Regionen in China)	Verpackungsalgorithmen: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (nur Regionen Chinas) Verpackungsschlüsselspezifikationen: RSA_2048 RSA_3072 RSA_4096 SM2 (nur China-Regionen)
HMAC-Schlüssel	Verpackungsalgorithmen: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Verpackungsschlüsselspezifikationen: RSA_2048 RSA_3072 RSA_4096

> **Note**
>
> Die Algorithmen `RSA_AES_KEY_WRAP_SHA_256` und `RSA_AES_KEY_WRAP_SHA_1` Wrapping werden in den Regionen Chinas nicht unterstützt.

- `RSA_AES_KEY_WRAP_SHA_256` – Ein zweistufiger Hybrid-Verpackungsalgorithmus, der die Verschlüsselung Ihres Schlüsselmaterials mit einem von Ihnen generierten symmetrischen AES-Schlüssel und die anschließende Verschlüsselung des symmetrischen AES-Schlüssels mit dem heruntergeladenen öffentlichen RSA-Verpackungsschlüssel und dem RSAES_OAEP_SHA_256-Verpackungsalgorithmus kombiniert.

  Für das `RSA_AES_KEY_WRAP_SHA_*` Verpacken von privatem RSA-Schlüsselmaterial ist ein Wrapping-Algorithmus erforderlich, außer in chinesischen Regionen, wo Sie den `SM2PKE` Wrapping-Algorithmus verwenden müssen.
- `RSA_AES_KEY_WRAP_SHA_1` – Ein zweistufiger Hybrid-Verpackungsalgorithmus, der die Verschlüsselung Ihres Schlüsselmaterials mit einem von Ihnen generierten symmetrischen AES-Schlüssel und die anschließende Verschlüsselung des symmetrischen AES-Schlüssels mit dem heruntergeladenen öffentlichen RSA-Verpackungsschlüssel und dem RSAES_OAEP_SHA_1-Verpackungsalgorithmus kombiniert.

  Für das `RSA_AES_KEY_WRAP_SHA_*` Verpacken von privatem RSA-Schlüsselmaterial ist ein Wrapping-Algorithmus erforderlich, außer in chinesischen Regionen, wo Sie den `SM2PKE` Wrapping-Algorithmus verwenden müssen.
- `RSAES_OAEP_SHA_256` – Der RSA-Verschlüsselungsalgorithmus mit Optimal Asymmetric Encryption Padding (OAEP) mit der SHA-256-Hash-Funktion.
- `RSAES_OAEP_SHA_1` – Der RSA-Verschlüsselungsalgorithmus mit Optimal Asymmetric Encryption Padding (OAEP) mit der SHA-1-Hash-Funktion.
- `RSAES_PKCS1_V1_5`(Veraltet; unterstützt seit dem 10. Oktober 2023 den Wrapping-Algorithmus RSAES_PKCS1_V1_5 AWS KMS nicht) — Der RSA-Verschlüsselungsalgorithmus mit dem in PKCS #1 Version 1.5 definierten Auffüllformat.
- `SM2PKE`(Nur Regionen Chinas) — Ein auf elliptischen Kurven basierender Verschlüsselungsalgorithmus, der von OSCCA in GM/T 0003.4-2012 definiert wurde.

Themen

- Herunterladen des öffentlichen Verpackungsschlüssels und Import-Tokens (Konsole)
- Der öffentliche Schlüssel für die Verpackung und das Import-Token (AWS KMS API) werden heruntergeladen

Herunterladen des öffentlichen Verpackungsschlüssels und Import-Tokens (Konsole)

Sie können die AWS KMS Konsole verwenden, um den öffentlichen Schlüssel für die Verpackung herunterzuladen und das Token zu importieren.

1. Wenn Sie gerade die Schritte zum Erstellen eines KMS-Schlüssel ohne Schlüsselmaterial abgeschlossen haben und sich auf der Seite Download wrapping key and import token (Umhüllungsschlüssel und Import-Token herunterladen) befinden, fahren Sie gleich mit Step 9 fort.
2. Melden Sie sich bei der Konsole AWS Key Management Service (AWS KMS) unter https://console.aws.amazon.com/kms an AWS Management Console und öffnen Sie sie.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.

   > **Tip**
   >
   > Sie können Schlüsselmaterial nur in einen KMS-Schlüssel mit dem Ursprung Extern (Schlüsselmaterial importieren). Dies bedeutet, dass der KMS-Schlüssel ohne Schlüsselmaterial erstellt wurde. Um die Spalte Origin (Ursprung) zu Ihrer Tabelle hinzuzufügen, klicken Sie auf das Symbol „Settings (Einstellungen)“ in der rechten oberen Ecke der Seite (). Aktivieren Sie Origin (Ursprung) und wählen Sie dann Confirm (Bestätigen) aus.

5. Wählen Sie den Alias oder die Schlüssel-ID des KMS-Schlüssels aus, dessen Import ausstehend ist.
6. Erweitern Sie den Bereich Cryptographic configuration (kryptografische Konfiguration) und zeigen Sie dessen Werte an. Die Registerkarte wird unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.

   Sie können Schlüsselmaterial nur in KMS-Schlüssel mit dem Ursprung Extern (Schlüsselmaterial importieren). Weitere Informationen zum Erstellen von KMS-Schlüsseln mit importiertem Schlüsselmaterial finden Sie unter Schlüsselmaterial für AWS KMS Schlüssel importieren.
7. Wählen Sie die Registerkarte Schlüsselmaterial und dann Schlüsselmaterial importieren.

Die Registerkarte Schlüsselmaterial wird nur für KMS-Schlüssel angezeigt, bei denen der Wert für Ursprung Extern (Schlüsselmaterial importieren) ist.

8. Wählen Sie unter Verpackungsschlüssel-Spezifikation auswählen die Konfiguration für Ihren KMS-Schlüssel aus. Nachdem Sie diesen Schlüssel erstellt haben, können Sie die Schlüsselspezifikationen nicht ändern.
9. Wählen Sie für Select wrapping algorithm die Option aus, die Sie zum Verschlüsseln Ihres Schlüsselmaterials verwenden werden. Weitere Informationen zu den Optionen finden Sie unter Verpackungsalgorithmus auswählen.
10. Klicken Sie auf Herunterladen des öffentlichen Verpackungsschlüssels und Import-Tokens und speichern Sie dann die Datei.

    Wenn die Option Next (Weiter) vorhanden ist, wählen Sie Next (Weiter), um den Vorgang fortzusetzen. Um später fortzufahren, wählen Sie Cancel (Abbrechen).
11. Dekomprimieren Sie die `.zip`-Datei, die Sie im vorherigen Schritt gespeichert haben (`Import_Parameters_<key_id>_<timestamp>`).

    Der Ordner enthält die folgenden Dateien:

    - Ein umschließender öffentlicher Schlüssel in eine Datei mit dem Namen. `WrappingPublicKey.bin`
    - Ein Import-Token in einer Datei mit dem Namen `ImportToken.bin`.
    - Eine Textdatei mit dem Namen README.txt. Diese Datei enthält Informationen über den öffentlichen Verpackungsschlüssel, den zu verwendenden Verpackungsschlüssel zum Verschlüsseln Ihres Schlüsselmaterials und das Datum und die Uhrzeit des Ablaufens des öffentlichen Verpackungsschlüssels und des Import-Tokens.
12. Um mit dem Vorgang fortzufahren, nehmen Sie ihn mit dem Schritt Verschlüsseln Ihres Schlüsselmaterials wieder auf.

Der öffentliche Schlüssel für die Verpackung und das Import-Token (AWS KMS API) werden heruntergeladen

Verwenden Sie die GetParametersForImportAPI, um den öffentlichen Schlüssel herunterzuladen und das Token zu importieren. Geben Sie den KMS-Schlüssel an, der mit dem importierten Schlüsselmaterial verknüpft werden soll. Dieser KMS-Schlüssel muss einen Ursprungs-Wert von`EXTERNAL` haben.

In diesem Beispiel werden der RSA_AES_KEY_WRAP_SHA_256-Verpackungsalgorithmus, die RSA_3072-Spezifikation zum Verpacken öffentlicher Schlüssel und ein Beispiel für eine Schlüssel-ID angegeben. Ersetzen Sie diese Beispielwerte durch gültige Werte für Ihren Download. Für die Schlüssel-ID können Sie eine Schlüssel-ID oder eine ARN des Schlüssels verwenden, aber keinen Aliasnamen oder Alias-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
```

Wenn der Befehl erfolgreich ausgeführt wurde, wird eine Ausgabe ähnlich der Folgenden angezeigt:

```
{
    "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)"
}
```

Um die Daten für den nächsten Schritt vorzubereiten, dekodiert base64 den öffentlichen Schlüssel und das Import-Token und speichert die dekodierten Werte in Dateien.

So dekodieren Sie den öffentlichen Schlüssel und das Import-Token von base64:

1. Kopieren Sie den base64-verschlüsselten öffentlichen Schlüssel (dargestellt durch *Öffentlicher Schlüssel (base64-verschlüsselt)* in der Beispielausgabe), fügen Sie sie in eine neue Datei ein und speichern Sie dann die Datei. Geben Sie der Datei einen aussagekräftigen Namen wie `PublicKey.b64`.
2. Verwenden Sie OpenSSL für die base64-Entschlüsselung der Inhalte der Datei und speichern Sie die entschlüsselten Daten in einer neuen Datei. Das folgende Beispiel decodiert die Daten in die Datei, die Sie im vorherigen Schritt gespeichert haben (`PublicKey.b64`), und speichert die Ausgabe in einer neuen Datei mit dem Namen `WrappingPublicKey.bin`.

   ```
   $ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
   ```
3. Kopieren Sie den base64-verschlüsselten Import-Token (dargestellt durch *Import-Token (base64-verschlüsselt)* in der Beispielausgabe), fügen Sie sie in eine neue Datei ein und

speichern Sie dann die Datei. Geben Sie der Datei einen aussagekräftigen Namen, wie z. B. `importtoken.b64`.

4. Verwenden Sie OpenSSL für die base64-Entschlüsselung der Inhalte der Datei und speichern Sie die entschlüsselten Daten in einer neuen Datei. Das folgende Beispiel decodiert die Daten in die Datei, die Sie im vorherigen Schritt gespeichert haben (`ImportToken.b64`), und speichert die Ausgabe in einer neuen Datei mit dem Namen `ImportToken.bin`.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Fahren Sie mit Schritt 3: Verschlüsselung des Schlüsselmaterials fort.

Schritt 3 für den Import von Schlüsselmaterial: Verschlüsselung des Schlüsselmaterials

Nachdem Sie den öffentlichen Schlüssel und das Import-Token heruntergeladen haben, verschlüsseln Sie Ihr Schlüsselmaterial mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben und dem von Ihnen angegebenen Verpackungsalgorithmus. Wenn Sie den öffentlichen Schlüssel oder das Import-Token ersetzen oder den Verpackungsalgorithmus ändern müssen, müssen Sie einen neuen öffentlichen Schlüssel und Import-Token herunterladen. Hinweise zu den AWS KMS unterstützten öffentlichen Schlüsseln und Wrapping-Algorithmen finden Sie unter Wählen Sie eine Spezifikation für den öffentlichen Verpackungsschlüssel und Auswählen des Verpackungsalgorithmus.

Das Schlüsselmaterial muss im Binärformat vorliegen. Weitere Informationen hierzu finden Sie unter Anforderungen an importiertes Schlüsselmaterial.

> **Note**
>
> Verschlüsseln und importieren Sie bei asymmetrischen Schlüsselpaaren nur den privaten Schlüssel. AWS KMS leitet den öffentlichen Schlüssel vom privaten Schlüssel ab.
> Die folgende Kombination wird NICHT unterstützt: ECC_NIST_P521-Schlüsselmaterial, die RSA_2048-Spezifikation für öffentliche Verpackungsschlüssel und ein RSAES_OAEP_SHA_*-Verpackungsalgorithmus.
> Sie können das Schlüsselmaterial von ECC_NIST_P521 nicht direkt mit einem öffentlichen RSA_2048-Verpackungsschlüssel verpacken. Verwenden Sie einen größeren Verpackungsschlüssel oder einen RSA_AES_KEY_WRAP_SHA_*-WRAP_*-Verpackungsalgorithmus.

Die Wrapping-Algorithmen RSA_AES_KEY_WRAP_SHA_256 und RSA_AES_KEY_WRAP_SHA_1 werden in chinesischen Regionen nicht unterstützt.

In der Regel verschlüsseln Sie das Schlüsselmaterial, wenn Sie es aus dem Hardwaresicherheitsmodul (HSM) oder Schlüsselverwaltungssystem exportieren. Weitere Informationen zum Exportieren von Schlüsselmaterial im Binärformat finden Sie in der Dokumentation zum HSM oder Schlüsselverwaltungssystem. Sie können sich auch den folgenden Abschnitt ansehen, der eine Machbarkeitsnachweis-Demonstration unter Verwendung von OpenSSL enthält.

Verwenden Sie beim Verschlüsseln Ihres Schlüsselmaterials den gleichen Verpackungsalgorithmus, den Sie beim Herunterladen des öffentlichen Schlüssels und Import-Tokens angegeben haben. Den von Ihnen angegebenen Wrapping-Algorithmus finden Sie im Protokollereignis für die zugehörige Anfrage. CloudTrail GetParametersForImport

Generieren Sie Schlüsselmaterial für Tests

Die folgenden OpenSSL-Befehle generieren Schlüsselmaterial für jeden unterstützten Typ zum Testen. Diese Beispiele dienen nur zu Test- und proof-of-concept Demonstrationszwecken. Verwenden Sie für Produktionssysteme eine sicherere Methode zur Generierung Ihres Schlüsselmaterials, z. B. ein Hardware-Sicherheitsmodul oder ein Schlüsselverwaltungssystem.

Um die privaten Schlüssel von asymmetrischen Schlüsselpaaren in das DER-kodierte Format zu konvertieren, leiten Sie den Befehl zur Erzeugung von Schlüsselmaterial an den folgenden `openssl pkcs8`-Befehl weiter. Der `topk8`-Parameter weist OpenSSL an, einen privaten Schlüssel als Eingabe zu verwenden und einen PKCS-#8-formatierten Schlüssel zurückzugeben. (Das Standardverhalten ist das Gegenteil.)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

Die folgenden Befehle erzeugen Testschlüsselmaterial für jeden der unterstützten Schlüsseltypen.

- Symmetrischer Verschlüsselungsschlüssel (32 Bytes)

  Dieser Befehl erzeugt einen symmetrischen 256-Bit-Schlüssel (32-Byte-Zufallszeichenfolge) und speichert ihn in der Datei `PlaintextKeyMaterial.bin`. Sie müssen dieses Schlüsselmaterial nicht verschlüsseln.

  ```
  openssl rand -out PlaintextKeyMaterial.bin 32
  ```

Nur in chinesischen Regionen müssen Sie einen symmetrischen 128-Bit-Schlüssel (16-Byte-Zufallszeichenfolge) generieren.

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- HMAC-Schlüssel

  Dieser Befehl generiert eine zufällige Byte-Zeichenfolge der angegebenen Größe. Sie müssen dieses Schlüsselmaterial nicht verschlüsseln.

  Die Länge Ihres HMAC-Schlüssels muss der Länge entsprechen, die in der Schlüsselspezifikation des KMS-Schlüssels definiert ist. Wenn der KMS-Schlüssel beispielsweise HMAC_384 lautet, müssen Sie einen 384-Bit-Schlüssel (48 Byte) importieren.

```
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
```

- RSA-Privatschlüssel

```
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
```

- ECC-Privatschlüssel

```
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
```

- Private SM2-Schlüssel (nur Regionen Chinas)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -
outform der -nocrypt > SM2_PrivateKey.der
```

Beispiele für die Verschlüsselung von Schlüsselmaterial mit OpenSSL

Die folgenden Beispiele zeigen, wie Sie OpenSSL verwenden, um Ihr Schlüsselmaterial mit dem öffentlichen Schlüssel zu verschlüsseln, den Sie heruntergeladen haben. Verwenden Sie die Klasse, um Ihr Schlüsselmaterial mit einem öffentlichen SM2-Schlüssel zu verschlüsseln (nur Regionen Chinas). `SM2OfflineOperationHelper`

> **Important**
>
> Diese Beispiele dienen nur zur Demonstration des Konzepts. Verwenden Sie bei Produktionssystemen eine sicherere Methode (beispielsweise ein kommerzielles HSM oder Schlüsselverwaltungssystem), um Ihr Schlüsselmaterial zu generieren und zu speichern.
> Die folgende Kombination wird NICHT unterstützt: ECC_NIST_P521-Schlüsselmaterial, die RSA_2048-Spezifikation für öffentliche Verpackungsschlüssel und ein RSAES_OAEP_SHA_*-Verpackungsalgorithmus.
> Sie können das Schlüsselmaterial von ECC_NIST_P521 nicht direkt mit einem öffentlichen RSA_2048-Verpackungsschlüssel verpacken. Verwenden Sie einen größeren Verpackungsschlüssel oder einen RSA_AES_KEY_WRAP_SHA_*-WRAP_*-Verpackungsalgorithmus.

RSAES_OAEP_SHA_1

AWS KMS unterstützt RSAES_OAEP_SHA_1 für symmetrische Verschlüsselungsschlüssel (SYMMETRIC_DEFAULT), private Schlüssel mit elliptischen Kurven (ECC), private SM2-Schlüssel und HMAC-Schlüssel.

RSAES_OAEP_SHA_1 wird für private RSA-Schlüssel nicht unterstützt. Außerdem können Sie einen öffentlichen RSA_2048-Verpackungsschlüssel nicht mit einem RSAES_OAEP_SHA_*-Verpackungsalgorithmus verwenden, um einen privaten ECC_NIST_P521 (secp521r1)-Schlüssel zu verpacken. Sie müssen einen größeren öffentlichen Verpackungsschlüssel oder einen RSA_AES_KEY_WRAP-Verpackungsalgorithmus verwenden.

Im folgenden Beispiel wird Ihr Schlüsselmaterial mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem Verpackungsalgorithmus RSAES_OAEP_SHA_1 verschlüsselt und in der Datei `EncryptedKeyMaterial.bin` gespeichert.

In diesem Beispiel:

- *`WrappingPublicKey.bin`* ist die Datei, die den heruntergeladenen öffentlichen Verpackungsschlüssel enthält.
- *`PlaintextKeyMaterial.bin`* ist die Datei, die das Schlüsselmaterial enthält, das Sie verschlüsseln, z. B. `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` oder `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 unterstützt RSAES_OAEP_SHA_256 für symmetrische Verschlüsselungsschlüssel (SYMMETRIC_DEFAULT), private Schlüssel mit elliptischen Kurven (ECC), private SM2-Schlüssel und HMAC-Schlüssel.

RSAES_OAEP_SHA_256 wird für private RSA-Schlüssel nicht unterstützt. Außerdem können Sie einen öffentlichen RSA_2048-Verpackungsschlüssel nicht mit einem RSAES_OAEP_SHA_*-Verpackungsalgorithmus verwenden, um einen privaten ECC_NIST_P521 (secp521r1)-Schlüssel zu verpacken. Sie müssen einen größeren öffentlichen Schlüssel oder einen RSA_AES_KEY_WRAP-Verpackungsalgorithmus verwenden.

Im folgenden Beispiel wird Schlüsselmaterial mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem Verpackungsalgorithmus RSAES_OAEP_SHA_256 verschlüsselt und in der Datei `EncryptedKeyMaterial.bin` gespeichert.

In diesem Beispiel:

- *`WrappingPublicKey.bin`* ist die Datei, die den heruntergeladenen öffentlichen Verpackungsschlüssel enthält. Wenn Sie den öffentlichen Schlüssel über die Konsole heruntergeladen haben, hat diese Datei den Namen `wrappingKey_`*`KMS key_key_ID_timestamp`* (z. B. `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *`PlaintextKeyMaterial.bin`* ist die Datei, die das Schlüsselmaterial enthält, das Sie verschlüsseln, z. B. `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` oder `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

Der Verpackungsalgorithmus RSA_AES_KEY_WRAP_SHA_1 umfasst zwei Verschlüsselungsoperationen.

1. Verpacken Sie Ihr Schlüsselmaterial mit einem von Ihnen generierten symmetrischen AES-Schlüssel und einem symmetrischen AES-Verpackungsalgorithmus.
2. Verschlüsseln Sie den symmetrischen AES-Schlüssel, den Sie verwendet haben, mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem RSAES_OAEP_SHA_1-Verpackungsalgorithmus.

AWS KMS unterstützt RSA_AES_KEY_WRAP_SHA_*-Wrap-Algorithmen für alle unterstützten Typen von importiertem Schlüsselmaterial und alle unterstützten Spezifikationen für öffentliche Schlüssel. Die RSA_AES_KEY_WRAP_SHA_*-Algorithmen sind die einzigen Verpackungsalgorithmen, die für das Verpacken von RSA-Schlüsselmaterial unterstützt werden.

Der Verpackungsalgorithmus RSA_AES_KEY_WRAP_SHA_1 erfordert OpenSSL Version 3. x oder später.

1. Generieren Sie einen symmetrischen 256-Bit-AES-Schlüssel für die Verschlüsselung

   Dieser Befehl generiert einen symmetrischen AES-Verpackungsschlüssel, der aus 256 zufälligen Bits besteht, und speichert ihn in der Datei `aes-key.bin`

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. Verschlüsseln Sie Ihr Schlüsselmaterial mit dem symmetrischen AES-Verschlüsselungsschlüssel

   Dieser Befehl verschlüsselt Ihr Schlüsselmaterial mit dem symmetrischen AES-Verschlüsselungsschlüssel und speichert das verschlüsselte Schlüsselmaterial in der Datei `key-material-wrapped.bin`.

   In diesem Beispielbefehl:

   - *`PlaintextKeyMaterial.bin`* ist die Datei, die das Schlüsselmaterial enthält, das Sie importieren, z. B. `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` oder `ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`* ist die Datei, die den symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel enthält, den Sie mit dem vorherigen Befehl erzeugt haben.

   ```
   # 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. Verschlüsseln Sie Ihren symmetrischen AES-Verschlüsselungsschlüssel mit dem öffentlichen Schlüssel

   Dieser Befehl verschlüsselt Ihren symmetrischen AES-Verschlüsselungsschlüssel mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem RSAES_OAEP_SHA_1-Verpackungsalgorithmus, DER-kodiert ihn und speichert ihn in der Datei `aes-key-wrapped.bin`.

   In diesem Beispielbefehl:

   - *`WrappingPublicKey.bin`* ist die Datei, die den heruntergeladenen öffentlichen Verpackungsschlüssel enthält. Wenn Sie den öffentlichen Schlüssel über die Konsole heruntergeladen haben, hat diese Datei den Namen `wrappingKey_`*`KMS key_key_ID_timestamp`* (z. B. `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`
   - *`aes-key.bin`* ist die Datei, die den symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel enthält, den Sie im ersten Befehl in dieser Beispielsequenz generiert haben.

   ```
   # 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. Generieren Sie die zu importierende Datei

   Verketten Sie die Datei mit dem verschlüsselten Schlüsselmaterial und die Datei mit dem verschlüsselten AES-Schlüssel. Speichern Sie sie in der `EncryptedKeyMaterial.bin`-Datei, die Sie in Schritt 4: Importieren des Schlüsselmaterials importieren werden.

   In diesem Beispielbefehl:

- *key-material-wrapped.bin* ist die Datei, die das verschlüsselte Schlüsselmaterial enthält.
- *aes-key-wrapped.bin* ist die Datei, die den verschlüsselten AES-Verschlüsselungsschlüssel enthält.

```
# 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

Der Verpackungsalgorithmus RSA_AES_KEY_WRAP_SHA_256 umfasst zwei Verschlüsselungsoperationen.

1. Verpacken Sie Ihr Schlüsselmaterial mit einem von Ihnen generierten symmetrischen AES-Schlüssel und einem symmetrischen AES-Verpackungsalgorithmus.
2. Verschlüsseln Sie den symmetrischen AES-Schlüssel, den Sie verwendet haben, mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem RSAES_OAEP_SHA_256-Verpackungsalgorithmus.

AWS KMS unterstützt RSA_AES_KEY_WRAP_SHA_*-Wrap-Algorithmen für alle unterstützten Typen von importiertem Schlüsselmaterial und alle unterstützten Spezifikationen für öffentliche Schlüssel. Die RSA_AES_KEY_WRAP_SHA_*-Algorithmen sind die einzigen Verpackungsalgorithmen, die für das Verpacken von RSA-Schlüsselmaterial unterstützt werden.

Der Verpackungsalgorithmus RSA_AES_KEY_WRAP_SHA_256 erfordert OpenSSL Version 3. x oder später.

1. Generieren Sie einen symmetrischen 256-Bit-AES-Schlüssel für die Verschlüsselung

   Dieser Befehl generiert einen symmetrischen AES-Verpackungsschlüssel, der aus 256 zufälligen Bits besteht, und speichert ihn in der Datei `aes-key.bin`

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. Verschlüsseln Sie Ihr Schlüsselmaterial mit dem symmetrischen AES-Verschlüsselungsschlüssel

   Dieser Befehl verschlüsselt Ihr Schlüsselmaterial mit dem symmetrischen AES-Verschlüsselungsschlüssel und speichert das verschlüsselte Schlüsselmaterial in der Datei `key-material-wrapped.bin`.

   In diesem Beispielbefehl:

   - *`PlaintextKeyMaterial.bin`* ist die Datei, die das Schlüsselmaterial enthält, das Sie importieren, z. B. `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` oder `ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`* ist die Datei, die den symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel enthält, den Sie mit dem vorherigen Befehl erzeugt haben.

   ```
   # 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. Verschlüsseln Sie Ihren symmetrischen AES-Verschlüsselungsschlüssel mit dem öffentlichen Schlüssel

   Dieser Befehl verschlüsselt Ihren symmetrischen AES-Verschlüsselungsschlüssel mit dem öffentlichen Schlüssel, den Sie heruntergeladen haben, und dem RSAES_OAEP_SHA_256-Verpackungsalgorithmus, DER-kodiert ihn und speichert ihn in der Datei `aes-key-wrapped.bin`.

   In diesem Beispielbefehl:

   - *`WrappingPublicKey.bin`* ist die Datei, die den heruntergeladenen öffentlichen Verpackungsschlüssel enthält. Wenn Sie den öffentlichen Schlüssel über die Konsole heruntergeladen haben, hat diese Datei den Namen `wrappingKey_`*`KMS key_key_ID_timestamp`* (z. B. `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`

- *aes-key.bin* ist die Datei, die den symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel enthält, den Sie im ersten Befehl in dieser Beispielsequenz generiert haben.

```
# 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. Generieren Sie die zu importierende Datei

   Verketten Sie die Datei mit dem verschlüsselten Schlüsselmaterial und die Datei mit dem verschlüsselten AES-Schlüssel. Speichern Sie sie in der `EncryptedKeyMaterial.bin`-Datei, die Sie in Schritt 4: Importieren des Schlüsselmaterials importieren werden.

   In diesem Beispielbefehl:

   - *key-material-wrapped.bin* ist die Datei, die das verschlüsselte Schlüsselmaterial enthält.
   - *aes-key-wrapped.bin* ist die Datei, die den verschlüsselten AES-Verschlüsselungsschlüssel enthält.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Fahren Sie mit Schritt 4: Importieren des Schlüsselmaterials fort.

Importieren von Schlüsselmaterial Schritt 4: Importieren des Schlüsselmaterials

Nachdem Sie Ihr Schlüsselmaterial verschlüsselt haben, können Sie das Schlüsselmaterial importieren, um es mit einem AWS KMS key zu verwenden. Um Schlüsselmaterial zu importieren, laden Sie das verschlüsselte Schlüsselmaterial aus Schritt 3: Verschlüsselung des Schlüsselmaterials und den Import-Token, den Sie von Schritt 2: Herunterladen des öffentlichen Verpackungsschlüssels und des Import-Tokens heruntergeladen haben, hoch. Sie müssen das Schlüsselmaterial in den gleichen KMS-Schlüssel importieren, den Sie auch beim Herunterladen des öffentlichen Schlüssels und Import-Tokens angegeben haben. Wenn Schlüsselmaterial erfolgreich importiert wird, ändert sich der Schlüsselstatus des KMS-Schlüssels auf `Enabled`, und Sie können den KMS-Schlüssel in kryptografischen Operationen verwenden.

Wenn Sie Schlüsselmaterial importieren, können Sie ein optionales Ablaufdatum für das Schlüsselmaterial festlegen. Wenn das Schlüsselmaterial abgelaufen ist, löscht AWS KMS das Schlüsselmaterial und der KMS-Schlüssel kann nicht mehr verwendet werden. Um den KMS-Schlüssel in kryptografischen Vorgängen zu verwenden, müssen Sie das gleiche Schlüsselmaterial erneut importieren. Nachdem Sie Ihr Schlüsselmaterial importiert haben, können Sie das Ablaufdatum für den aktuellen Import nicht mehr festlegen, ändern oder stornieren. Um diese Werte zu ändern, müssen Sie das gleiche Schlüsselmaterial löschen und wieder importieren.

Um Schlüsselmaterial zu importieren, können Sie die -AWS KMSKonsole oder die ImportKeyMaterial-API verwenden. Sie können die API direkt mittels HTTP-Anforderungen verwenden oder mithilfe eines AWS-SDKs, AWS Command Line Interface oder AWS Tools for PowerShell.

Wenn Sie das Schlüsselmaterial importieren, wird Ihrem AWS CloudTrail Protokoll ein ImportKeyMaterial Eintrag hinzugefügt, um den `ImportKeyMaterial` Vorgang aufzuzeichnen. Der CloudTrail Eintrag ist derselbe, unabhängig davon, ob Sie die AWS KMS Konsole oder die AWS KMS API verwenden.

Festlegen einer Ablaufzeit (optional)

Wenn Sie das Schlüsselmaterial für Ihren KMS-Schlüssel importieren, können Sie optional ein Ablaufdatum und eine Ablaufzeit für das Schlüsselmaterial von bis zu 365 Tagen ab dem Importdatum festlegen. Wenn importiertes Schlüsselmaterial abläuft, löscht AWS KMS es. Diese Aktion ändert den Status des KMS-Schlüssels zu `PendingImport`, wodurch verhindert wird, dass er in kryptografischen Operationen verwendet wird. Wenn Sie den KMS-Schlüssel verwenden möchten, müssen Sie das Schlüsselmaterial erneut importieren.

Wenn Sie sicherstellen, dass importiertes Schlüsselmaterial häufig abläuft, können Sie zwar die gesetzlichen Anforderungen erfüllen, jedoch birgt dies ein zusätzliches Risiko für Daten, die unter dem KMS-Schlüssel verschlüsselt sind. Solange Sie keine Kopie des ursprünglichen Schlüsselmaterials neu importieren, ist ein KMS-Schlüssel mit abgelaufenem Schlüsselmaterial unbrauchbar, und alle unter dem KMS-Schlüssel verschlüsselten Daten sind unzugänglich. Wenn Sie das Schlüsselmaterial aus irgendeinem Grund, einschließlich des Verlusts Ihrer Kopie des ursprünglichen Schlüsselmaterials, nicht erneut importieren, ist der KMS-Schlüssel dauerhaft unbrauchbar, und die unter dem KMS-Schlüssel verschlüsselten Daten sind nicht wiederherstellbar.

Um dieses Risiko zu mindern, stellen Sie sicher, dass Ihre Kopie des importierten Schlüsselmaterials zugänglich ist, und entwickeln Sie ein System, mit dem Sie das Schlüsselmaterial löschen und erneut importieren können, bevor es abläuft und Ihren AWS-Workload unterbricht. Wir empfehlen Ihnen, einen Alarm für den Ablauf Ihres importierten Schlüsselmaterials einzustellen, der Ihnen genügend Zeit gibt, das Schlüsselmaterial vor Ablauf erneut zu importieren. Sie können Ihre - CloudTrail Protokolle auch verwenden, um Vorgänge zu prüfen, die Schlüsselmaterial importieren (und erneut importieren) und importiertes Schlüsselmaterial löschen, sowie den AWS KMS Vorgang zum Löschen abgelaufenen Schlüsselmaterials.

Sie können kein anderes Schlüsselmaterial in den KMS-Schlüssel importieren, und AWS KMS kann das gelöschte Schlüsselmaterial nicht wiederherstellen oder reproduzieren. Anstatt eine Ablaufzeit festzulegen, können Sie das importierte Schlüsselmaterial programmgesteuert löschen und in regelmäßigen Abständen erneut importieren. Die Anforderungen für die Beibehaltung einer Kopie des ursprünglichen Schlüsselmaterials sind jedoch dieselben.

Sie legen fest, ob und wann importiertes Schlüsselmaterial abläuft, wenn Sie das Schlüsselmaterial importieren. Sie können das Ablaufdatum jedoch ein- und ausschalten oder eine neue Ablaufzeit festlegen, indem Sie das Schlüsselmaterial löschen und erneut importieren. Verwenden Sie den `ExpirationModel` Parameter ImportKeyMaterial, um den Ablauf zu aktivieren (`KEY_MATERIAL_EXPIRES`) und aus (`KEY_MATERIAL_DOES_NOT_EXPIRE`) und den `ValidTo` Parameter , um die Ablaufzeit festzulegen. Der maximale Zeitraum beträgt 365 Tage ab dem Import der Daten. Es gibt kein Minimum, aber der Zeitpunkt muss in der Zukunft liegen.

Importieren von Schlüsselmaterial (Konsole)

Sie können die AWS Management Console verwenden, um Schlüsselmaterial zu importieren.

1. Wenn Sie sich auf der Seite Ihr verpacktes Schlüsselmaterial hochladen befinden, fahren Sie mit Step 8 fort.

2. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
3. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
5. Wählen Sie die Schlüssel-ID oder den Alias des KMS-Schlüssels, für den Sie den öffentlichen Schlüssel und das Import-Token heruntergeladen haben.
6. Erweitern Sie den Bereich Cryptographic configuration (kryptografische Konfiguration) und zeigen Sie dessen Werte an. Die Registerkarten befinden sich auf der Detailseite für einen KMS-Schlüssel unter dem Abschnitt General configuration (allgemeine Konfiguration).

   Sie können Schlüsselmaterial nur in KMS-Schlüssel mit dem Ursprung Extern (Schlüsselmaterial importieren). Weitere Informationen zum Erstellen eines KMS-Schlüssels mit importiertem Schlüsselmaterial finden Sie unter Schlüsselmaterial für AWS KMS Schlüssel importieren.
7. Wählen Sie die Registerkarte Schlüsselmaterial und dann Schlüsselmaterial importieren. Die Registerkarte Schlüsselmaterial wird nur für KMS-Schlüssel angezeigt, bei denen der Wert für Ursprung Extern (Schlüsselmaterial importieren) ist.

   Wenn Sie das Schlüsselmaterial und Import-Token heruntergeladen haben und das Schlüsselmaterial verschlüsselt haben, wählen Sie Weiter.
8. Gehen Sie im Abschnitt Verschlüsseltes Schlüsselmaterial und Import-Token wie folgt vor.

   a. Wählen Sie unter Verpacktes Schlüsselmaterial Datei auswählen aus. Laden Sie dann die Datei hoch, die das durch den Umhüllungsschlüssel geschützte Schlüsselmaterial enthält.

   b. Wählen Sie unter Token importieren die Option Datei wählen. Laden Sie die heruntergeladene Datei mit dem Import-Token hoch.
9. Wählen Sie im Bereich Expiration option (Ablaufoption) aus, ob das Schlüsselmaterial abläuft. Um ein Ablaufdatum und eine Uhrzeit festzulegen, wählen Sie Key material expires (Schlüsselmaterial läuft ab) und wählen Sie dann Datum und Uhrzeit über das Kalendersteuerelement aus. Sie können ein Datum angeben, das bis zu 365 Tage nach dem aktuellen Datum und der aktuellen Uhrzeit liegt.
10. Wählen Sie die Option Upload key material (Schlüsselmaterial hochladen).

Schlüsselmaterial importieren (AWS KMS-API)

Um Schlüsselmaterial zu importieren, verwenden Sie die -ImportKeyMaterialOperation. Für die folgenden Beispiele wird die AWS CLIverwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Zur Verwendung dieses Beispiels gehen Sie wie folgt vor:

1. Ersetzen Sie *1234abcd-12ab-34cd-56ef-1234567890ab* durch eine Schlüssel-ID des KMS-Schlüssels, den Sie beim Herunterladen des öffentlichen Schlüssels und des Import-Tokens angegeben haben. Verwenden Sie zum Identifizieren des KMS-Schlüssels seine Schlüssel-ID oder den Schlüssel-ARN. Sie können für diese Operation keinen Alias-Namen oder Alias-ARN verwenden.
2. Ersetzen Sie *EncryptedKeyMaterial.bin* durch den Namen der Datei, die das verschlüsselten Schlüsselmaterial enthält.
3. Ersetzen Sie *ImportToken.bin* durch den Namen der Datei, die den Import-Token enthält.
4. Wenn das importierte Schlüsselmaterial ablaufen soll, setzen Sie den Wert für den `expiration-model`-Parameter auf den Standardwert zurück, `KEY_MATERIAL_EXPIRES`, oder lassen Sie den `expiration-model`-Parameter weg. Ersetzen Sie dann den Wert des `valid-to`-Parameters mit dem Datum und der Uhrzeit, zu der das Schlüsselmaterial ablaufen soll. Datum und Uhrzeit können bis zu 365 Tage ab dem Zeitpunkt der Anfrage betragen.

```
$ 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
```

   Wenn das importierte Schlüsselmaterial nicht ablaufen soll, setzen Sie den Wert für den `expiration-model`-Parameter auf`KEY_MATERIAL_DOES_NOT_EXPIRE` und lassen Sie den `valid-to`-Parameter aus dem Befehl weg.

```
$ 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**
>
> Wenn der Befehl nicht erfolgreich ist, wird möglicherweise eine `KMSInvalidStateException` oder eine `NotFoundException`angezeigt. Sie können die Anfrage wiederholen.

Benutzerdefinierte Schlüsselspeicher

Ein Schlüsselspeicher ist ein sicherer Ort zum Speichern kryptographischer Schlüssel. Der Standard-Schlüsselspeicher in AWS KMS unterstützt auch Methoden zum Generieren und Verwalten der Schlüssel, die er speichert. Standardmäßig wird das kryptografische Schlüsselmaterial für die AWS KMS keys, die Sie in AWS KMS erstellen, in Hardware-Sicherheitsmodulen (HSMs) generiert und geschützt, bei denen es sich um FIPS-140-2-validierte kryptografische Module handelt. Schlüsselmaterial für Ihre KMS-Schlüssel verlässt zu keiner Zeit Ihr HSMs unverschlüsselt.

Wenn Sie jedoch noch mehr Kontrolle über die HSMs benötigen, können Sie einen benutzerdefinierten Schlüsselspeicher erstellen.

Ein benutzerdefinierter Schlüsselspeicher ist ein logischer Schlüsselspeicher innerhalb von AWS KMS, der von einem Schlüsselmanager außerhalb von AWS KMS gestützt wird und Ihnen gehört und von Ihnen verwaltet wird. Benutzerdefinierte Schlüsselspeicher verbinden die benutzerfreundliche und umfassende Schlüsselverwaltungs-Oberfläche von AWS KMS mit der Fähigkeit, das Schlüsselmaterial und die kryptografischen Vorgänge zu besitzen und zu kontrollieren. Wenn Sie einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden, werden die kryptografischen Vorgänge tatsächlich von Ihrem Schlüsselmanager unter Verwendung Ihrer kryptografischen Schlüssel durchgeführt. Dadurch übernehmen Sie mehr Verantwortung für die Verfügbarkeit und Haltbarkeit kryptografischer Schlüssel sowie für den Betrieb der HSMs.

AWS KMS unterstützt zwei Arten von benutzerdefinierten Schlüsselspeichern.

- Bei einem AWS CloudHSM-Schlüsselspeicher handelt es sich um einen benutzerdefinierten AWS KMS-Schlüsselspeicher, der von einem AWS CloudHSM-Cluster unterstützt wird. Wenn Sie einen KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher speichern, generiert AWS KMS einen persistenten und nicht exportierbaren 256-Bit-Schlüssel nach Advanced Encryption Standard (AES) in dem zugeordneten AWS CloudHSM-Cluster. Dieses Schlüsselmaterial verlässt zu keiner Zeit Ihre AWS CloudHSM-Cluster unverschlüsselt. Wenn Sie einen KMS-Schlüssel im AWS CloudHSM-Schlüsselspeicher verwenden, werden die kryptografischen Vorgänge in den

HSMs im Cluster ausgeführt. AWS CloudHSM-Cluster werden von Hardware-Sicherheitsmodulen (HSMs) gestützt, die nach FIPS 140-2 Level 3 zertifiziert sind.

- Ein externer Schlüsselspeicher ist ein benutzerdefinierter AWS KMS-Schlüsselspeicher, der von einem externen Schlüsselmanager außerhalb von AWS unterstützt wird, den Sie besitzen und kontrollieren. Wenn Sie einen KMS-Schlüssel in Ihrem externen Schlüsselspeicher verwenden, werden die Ver- und Entschlüsselungsvorgänge von Ihrem externen Schlüsselmanager unter Verwendung Ihrer kryptographischen Schlüssel durchgeführt. Externe Schlüsselspeicher sind so konzipiert, dass sie eine Vielzahl von externen Schlüsselmanagern verschiedener Anbieter unterstützen.

  AWS KMS zeigt Ihren externen Schlüsselmanager oder Ihre kryptografischen Schlüssel niemals direkt an, greift direkt auf sie zu oder interagiert direkt mit ihnen. Wenn Sie mit einem KMS-Schlüssel in Ihrem externen Schlüsselspeicher verschlüsseln oder entschlüsseln, wird der Vorgang von Ihrem externen Schlüsselmanager unter Verwendung Ihrer externen Schlüssel durchgeführt. Sie behalten die volle Kontrolle über Ihre kryptografischen Schlüssel, einschließlich der Möglichkeit, einen kryptografischen Vorgang abzulehnen oder zu stoppen, ohne mit AWS zu interagieren. Aufgrund der Entfernung und der zusätzlichen Verarbeitung weisen KMS-Schlüssel in einem externen Schlüsselspeicher jedoch möglicherweise eine schlechtere Latenz und Leistung sowie andere Verfügbarkeitsmerkmale als KMS-Schlüssel mit Schlüsselmaterial in AWS KMS auf. Weitere Informationen zu Schlüsselmanagern, die mit dem externen Schlüsselspeicher-Feature AWS KMS kompatibel sind, finden Sie unter Welche externen Anbieter unterstützen die XKS-Proxy-Spezifikation? in den AWS Key Management Service-FAQs.

Diese beiden Arten von benutzerdefinierten Schlüsselspeichern unterscheiden sich erheblich vom Standardschlüsselspeicher AWS KMS und voneinander. Ihre Sicherheitsmodelle, ihr Verantwortungsbereich, ihre Leistung, ihr Preis und die Anwendungsfälle unterscheiden sich ebenfalls erheblich. Bevor Sie sich für einen benutzerdefinierten Schlüsselspeicher entscheiden, lesen Sie die zugehörige Dokumentation und vergewissern Sie sich, dass die zusätzliche Verantwortung für Konfiguration und Wartung ein sinnvoller Kompromiss für die zusätzliche Kontrolle ist. Wenn die Regeln und Vorschriften, nach denen Sie arbeiten, jedoch eine direkte Kontrolle über das Schlüsselmaterial vorschreiben, ist ein benutzerdefinierter Schlüsselspeicher möglicherweise sinnvoll für Sie.

Nicht unterstützte Funktionen

AWS KMS unterstützt die folgenden Funktionen in benutzerdefinierten Schlüsselspeichern nicht.

- Asymmetrische KMS-Schlüssel

- Asymmetrische Datenschlüsselpaare
- HMAC-KMS-Schlüssel
- KMS-Schlüssel mit importiertem Schlüsselmaterial
- Automatische Schlüsselrotation
- Multiregionale Schlüssel

Themen

- AWS CloudHSM wichtige Geschäfte
- Externe Schlüsselspeicher

AWS CloudHSM wichtige Geschäfte

Ein AWS CloudHSM Schlüsselspeicher ist ein benutzerdefinierter Schlüsselspeicher, der von einem AWS CloudHSM Cluster unterstützt wird. Wenn Sie einen AWS KMS keyin einem benutzerdefinierten Schlüsselspeicher erstellen, AWS KMS generiert und speichert er nicht extrahierbares Schlüsselmaterial für den KMS-Schlüssel in einem AWS CloudHSM Cluster, den Sie besitzen und verwalten. Wenn Sie einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden, werden die kryptografischen Operationen in den HSMs im Cluster ausgeführt. Diese Funktion kombiniert den Komfort und die umfassende Integration von AWS KMS mit der zusätzlichen Steuerung eines AWS CloudHSM Clusters in Ihrem. AWS-Konto

AWS KMS bietet vollständige Konsolen- und API-Unterstützung für die Erstellung, Verwendung und Verwaltung Ihrer benutzerdefinierten Schlüsselspeicher. Verwenden Sie die KMS-Schlüssel in Ihrem benutzerdefinierten Schlüsselspeicher auf die gleiche Weise, wie Sie jeden KMS-Schlüssel verwenden. Sie können KMS-Schlüssel beispielsweise zum Generieren von Datenschlüsseln und zum Verschlüsseln von Daten verwenden. Sie können die KMS-Schlüssel auch in Ihrem benutzerdefinierten Schlüsselspeicher mit AWS Diensten verwenden, die vom Kunden verwaltete Schlüssel unterstützen.

Brauche ich einen benutzerdefinierten Schlüsselspeicher?

Für die meisten Benutzer erfüllt der AWS KMS Standard-Schlüsselspeicher, der durch FIPS 140-2-validierte kryptografische Module geschützt ist, ihre Sicherheitsanforderungen. Es gibt keinen besonderen Grund, eine zusätzliche Ebene von Wartungsverantwortlichkeit und eine Abhängigkeit zu einem weiteren Service zu schaffen.

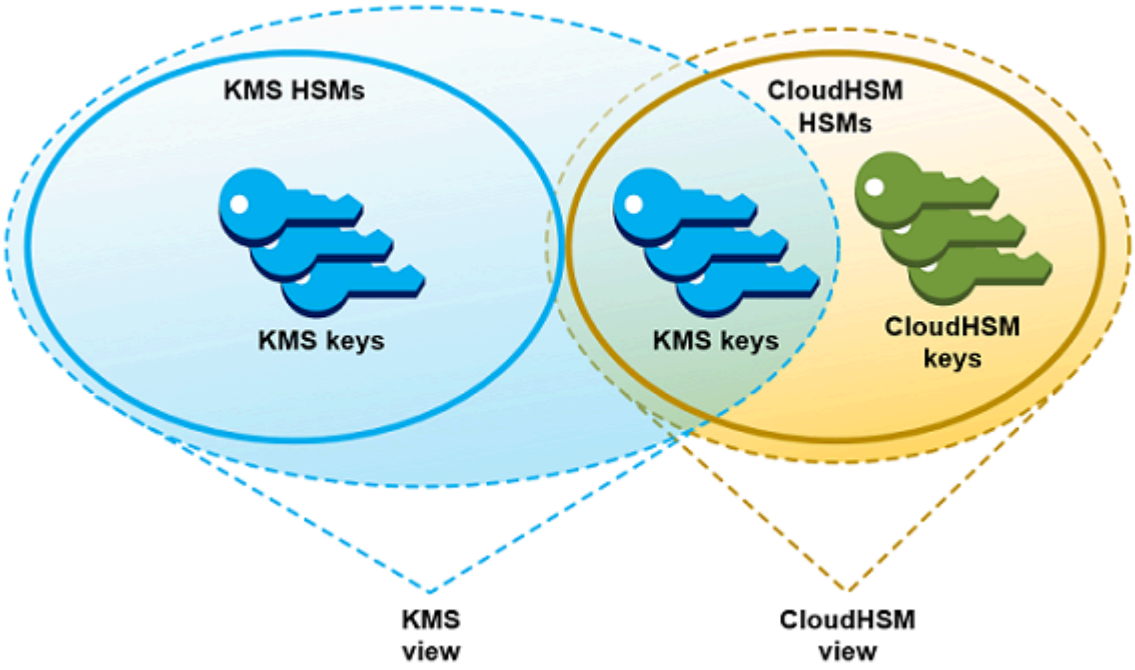
Es könnte jedoch für Sie sinnvoll sein, einen benutzerdefinierten Schlüsselspeicher zu erstellen, wenn für Sie eine der folgenden Anforderungen gilt:

- Sie haben Schlüssel, die ausdrücklich geschützt werden müssen, in einem Einzelmandanten-HSM oder in einem HSM, über das Sie direkte Kontrolle haben.
- Sie müssen in der Lage sein, Schlüsselmaterial sofort von zu entfernen. AWS KMS
- Sie müssen in der Lage sein, die gesamte Verwendung Ihrer Schlüssel unabhängig von AWS KMS oder zu überprüfen AWS CloudTrail.

Wie funktionieren benutzerdefinierten Schlüsselspeicher?

Jeder benutzerdefinierte Schlüsselspeicher ist einem AWS CloudHSM Cluster in Ihrem zugeordnet AWS-Konto. Wenn Sie den benutzerdefinierten Schlüsselspeicher mit seinem Cluster verbinden, AWS KMS wird die Netzwerkinfrastruktur zur Unterstützung der Verbindung erstellt. Anschließend meldet es sich mit den Anmeldeinformationen eines dedizierten Krypto-Benutzers im Cluster beim AWS CloudHSM Key-Client im Cluster an.

Sie erstellen und verwalten Ihre benutzerdefinierten Schlüsselspeicher AWS KMS und erstellen und verwalten Ihre HSM-Cluster in AWS CloudHSM. Wenn Sie einen AWS KMS benutzerdefinierten Schlüsselspeicher erstellen AWS KMS keys , können Sie die KMS-Schlüssel in AWS KMS anzeigen und verwalten. Sie können ihr Schlüsselmaterial aber auch dort anzeigen und verwalten AWS CloudHSM, genau wie Sie es für andere Schlüssel im Cluster tun würden.



Sie können KMS-Schlüssel mit symmetrischer Verschlüsselung mit Schlüsselmaterial erstellen, das AWS KMS in Ihrem benutzerdefinierten Schlüsselspeicher generiert wurde. Verwenden Sie dann dieselben Techniken, um die KMS-Schlüssel in Ihrem benutzerdefinierten Schlüsselspeicher anzuzeigen und zu verwalten, die Sie für KMS-Schlüssel im AWS KMS Schlüsselspeicher verwenden. Sie können den Zugriff über IAM und Schlüsselrichtlinien steuern, Tags und Aliasse erstellen, die KMS-Schlüssel aktivieren und deaktivieren und die Löschung von Schlüsseln planen. Sie können die KMS-Schlüssel für kryptografische Operationen verwenden und sie mit AWS Diensten verwenden, die sich in integrieren lassen. AWS KMS

Darüber hinaus haben Sie die volle Kontrolle über den AWS CloudHSM Cluster, einschließlich der Erstellung und Löschung von HSMs und der Verwaltung von Backups. Sie können den AWS CloudHSM Client und die unterstützten Softwarebibliotheken verwenden, um das Schlüsselmaterial für Ihre KMS-Schlüssel einzusehen, zu prüfen und zu verwalten. Solange der benutzerdefinierte Schlüsselspeicher getrennt ist, AWS KMS kann er nicht darauf zugreifen, und Benutzer können die KMS-Schlüssel im benutzerdefinierten Schlüsselspeicher nicht für kryptografische Operationen verwenden. Diese zusätzliche Kontrollebene macht benutzerdefinierte Schlüsselspeicher zu einer leistungsstarken Lösung für Unternehmen mit entsprechenden Anforderungen.

Wo fange ich an?

Um einen AWS CloudHSM Schlüsselspeicher zu erstellen und zu verwalten, verwenden Sie die Funktionen von AWS KMS und. AWS CloudHSM

1. Fangen Sie an AWS CloudHSM. Erstellen Sie einen aktiven AWS CloudHSM -Cluster oder wählen Sie einen bestehenden aus. Der Cluster muss mindestens zwei aktive HSMs in verschiedenen Availability Zones haben. Erstellen Sie dann ein dediziertes Krypto-Benutzerkonto (CU-Konto) für AWS KMS in diesem Cluster.
2. Erstellen Sie unter einen benutzerdefinierten Schlüsselspeicher, der Ihrem ausgewählten AWS CloudHSM Cluster zugeordnet ist. AWS KMS AWS KMS bietet eine vollständige Verwaltungsoberfläche, mit der Sie Ihre benutzerdefinierten Schlüsselspeicher erstellen, anzeigen, bearbeiten und löschen können.
3. Wenn Sie bereit sind, Ihren benutzerdefinierten Schlüsselspeicher zu verwenden, verbinden Sie ihn mit dem zugehörigen AWS CloudHSM Cluster. AWS KMS erstellt die Netzwerkinfrastruktur, die zur Unterstützung der Verbindung benötigt wird. Anschließend meldet sich KMS mit den Anmeldeinformationen des dedizierten Krypto-Benutzers an dem Cluster an, um das Schlüsselmaterial in dem Cluster zu generieren und zu verwalten.
4. Jetzt können Sie KMS-Schlüssel mit symmetrischer Verschlüsselung in Ihrem benutzerdefinierten Schlüsselspeicher erstellen. Sie brauchen dazu lediglich bei der Erstellung eines KMS-Schlüssels Ihren benutzerdefinierten Schlüsselspeicher anzugeben.

Falls Sie bei dem Prozess Hilfe benötigen, schlagen Sie im Thema Fehlerbehebung für einen Custom Key Store nach. Wenn Sie dort keine Antwort oder Lösung finden, Verwenden Sie den Feedback-Link unten auf der Seite in diesem Handbuch, oder posten Sie eine Frage im AWS Key Management Service -Diskussionsforum.

Kontingente

AWS KMS ermöglicht bis zu 10 benutzerdefinierte Schlüsselspeicher in jeder AWS-Konto Region, einschließlich sowohl AWS CloudHSM Schlüsselspeichern als auch externer Schlüsselspeicher, unabhängig von deren Verbindungsstatus. Darüber hinaus gibt es AWS KMS Anforderungskontingente für die Verwendung von KMS-Schlüsseln in einem AWS CloudHSM Schlüsselspeicher.

Preise

Informationen zu den Kosten für AWS KMS benutzerdefinierte Schlüsselspeicher und vom Kunden verwaltete Schlüssel in einem benutzerdefinierten Schlüsselspeicher finden Sie unter AWS Key Management Service Preise. Informationen zu den Kosten von AWS CloudHSM Clustern und HSMs finden Sie unter AWS CloudHSM Preise.

Regionen

AWS KMS unterstützt AWS CloudHSM wichtige Geschäfte in allen Ländern, in AWS-Regionen denen dies unterstützt AWS KMS wird, mit Ausnahme von Asien-Pazifik (Melbourne), China (Peking), China (Ningxia) und Europa (Spanien).

Nicht unterstützte Funktionen

AWS KMS unterstützt die folgenden Funktionen in benutzerdefinierten Schlüsselspeichern nicht.

- Asymmetrische KMS-Schlüssel
- Asymmetrische Datenschlüsselpaare
- HMAC-KMS-Schlüssel
- KMS-Schlüssel mit importiertem Schlüsselmaterial
- Automatische Schlüsselrotation
- Multiregionale Schlüssel

Themen

Wichtige Konzepte für AWS CloudHSM-Schlüsselspeicher

In diesem Thema erläutern wir einige der Konzepte in AWS CloudHSM-Schlüsselspeichern.

AWS CloudHSM-Schlüsselspeicher

Ein AWS CloudHSM-Schlüsselspeicher ist ein benutzerdefinierter Schlüsselspeicher, der einemAWS CloudHSM-Cluster zugeordnet ist, den Sie besitzen und verwalten. AWS CloudHSM-Cluster werden durch FIPS 140-2 Level 3-konforme Hardware-Sicherheitsmodule (HSMs) gestützt.

Wenn Sie einen KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher speichern, generiert AWS KMS einen persistenten und nicht exportierbaren 256-Bit-Schlüssel nach Advanced Encryption Standard (AES) in dem zugeordneten AWS CloudHSM-Cluster. Dieses Schlüsselmaterial verlässt

zu keiner Zeit Ihr HSMs unverschlüsselt. Wenn Sie einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher verwenden, werden die kryptografischen Vorgänge in den HSMs im Cluster ausgeführt.

AWS CloudHSM-Schlüsselspeicher verbinden die benutzerfreundliche und umfassende Schlüsselverwaltungs-Oberfläche von AWS KMS mit den zusätzlichen Steuerungsmöglichkeiten eines AWS CloudHSM-Clusters in Ihrem AWS-Konto. Mit dieser integrierten Funktion können Sie KMS-Schlüssel in AWS KMS erstellen, verwalten und verwenden, wobei Sie die vollständige Kontrolle der HSMs wahren, die ihr Schlüsselmaterial speichern, darunter Managing-Cluster, HSMs und Backups. Sie können den AWS CloudHSM-Schlüsselspeicher und seine KMS-Schlüssel mit der AWS KMS-Konsole und APIs verwalten. Sie können auch die AWS CloudHSM-Konsole, APIs, Client-Software und zugehörige Softwarebibliotheken verwenden, um den zugehörigen Cluster zu verwalten.

Sie können Ihren AWS CloudHSM-Schlüsselspeicher anzeigen und verwalten, seine Eigenschaften bearbeiten und seine zugehörigen AWS CloudHSM-Cluster verbinden und trennen. Wenn Sie einen AWS CloudHSM-Schlüsselspeicher löschen, müssen Sie zuerst die KMS-Schlüssel in dem AWS CloudHSM-Schlüsselspeicher löschen, indem Sie deren Löschung planen und warten, bis die Wartezeit verstrichen ist. Das Löschen der AWS CloudHSM-Schlüsselspeicher entfernt die Ressource aus AWS KMS, wirkt sich jedoch nicht auf Ihren AWS CloudHSM-Cluster aus.

AWS CloudHSM-Cluster

Jeder AWS CloudHSM-Schlüsselspeicher ist einem AWS CloudHSM-Cluster zugeordnet. Wenn Sie einen AWS KMS key in Ihrem AWS CloudHSM-Schlüsselspeicher erstellen, erstellt AWS KMS dessen Schlüsselmaterial in dem zugehörigen Cluster. Wenn Sie einen KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher verwenden, wird die kryptographische Produktion in dem zugehörigen Cluster ausgeführt.

Jeder AWS CloudHSM-Cluster kann nur mit einem AWS CloudHSM-Schlüsselspeicher verbunden werden. Der Cluster, den Sie auswählen, kann nicht mit einem anderen AWS CloudHSM-Schlüsselspeicher assoziiert werden oder einen gemeinsamen Sicherungsverlauf mit einem zugehörigen AWS CloudHSM-Cluster haben. Der Cluster muss initialisiert und aktiv sein, und er muss sich in demselben AWS-Konto und in derselben Region wie der AWS CloudHSM-Schlüsselspeicher befinden. Sie können einen neuen Cluster erstellen oder einen vorhandenen verwenden. AWS KMS benötigt keinen exklusiven Zugriff auf den Cluster. Um KMS-Schlüssel in dem AWS CloudHSM-Schlüsselspeicher erstellen zu können, muss der zugehörige Cluster mindestens zwei aktive HSMs enthalten. Alle anderen Operationen erfordern nur ein HSM.

Sie geben den AWS CloudHSM-Cluster beim Erstellen des AWS CloudHSM-Schlüsselspeichers an; dieser kann nicht geändert werden. Sie können jedoch eine Cluster als Ersatz verwenden, der den gleichen Sicherungsverlauf wie der ursprüngliche Cluster hat. So können Sie bei Bedarf den Cluster löschen und ihn durch einen aus einer seiner Sicherungen erstellten Cluster ersetzen. Sie behalten die vollständige Kontrolle über den zugehörigen AWS CloudHSM-Cluster und können daher Benutzer und Schlüssel verwalten, HSMs erstellen und löschen sowie Sicherungen verwenden und verwalten.

Wenn Sie bereit sind, Ihren AWS CloudHSM-Schlüsselspeicher zu verwenden, speichern Sie ihn in dem zugehörigen AWS CloudHSM-Cluster. Sie können Ihren Custom Key Store jederzeit verbinden oder trennen. Wenn ein benutzerdefinierten Schlüsselspeicher verbunden ist, können Sie seine KMS-Schlüssel erstellen und verwenden. Wenn er getrennt ist, können Sie den benutzerdefinierten AWS CloudHSM-Schlüsselspeicher und seine KMS-Schlüssel anzeigen und verwalten. Sie können jedoch keine neuen KMS-Schlüssel erstellen oder die KMS-Schlüssel in dem AWS CloudHSM-Schlüsselspeicher für kryptographische Operationen verwenden.

`kmsuser`-Kryptobenutzer

Zum Erstellen und Verwalten von Schlüsselmaterial in dem zugehörigen AWS CloudHSM-Cluster für Sie verwendet AWS KMS einen dedizierten AWS CloudHSM-Kryptobenutzer (CU, Crypto User) in dem Cluster mit der Bezeichnung `kmsuser`. Der `kmsuser`-CU ist ein Standard-CU-Konto, das automatisch mit allen HSMs in dem Cluster synchronisiert und in Cluster-Sicherungen gespeichert wird.

Bevor Sie Ihren AWS CloudHSM-Schlüsselspeicher erstellen, erstellen Sie ein `kmsuser`-CU-Konto in Ihrem AWS CloudHSM-Cluster mit dem Befehl createUser in cloudhsm_mgmt_util. Wenn Sie dann den AWS CloudHSM-Schlüsselspeicher erstellen, geben Sie das `kmsuser`-Kontopasswort für AWS KMS an. Wenn Sie den benutzerdefinierten Schlüsselspeicher verbinden, meldet sich AWS KMS im Cluster als der `kmsuser`-CU an und rotiert sein Passwort. AWS KMS verschlüsselt das `kmsuser`-Passwort und speichert es dann sicher ab. Wenn das Passwort rotiert wird, wird das neue Passwort verschlüsselt und auf die gleiche Weise gespeichert.

AWS KMS bleibt als `kmsuser` angemeldet, solange der AWS CloudHSM-Schlüsselspeicher verbunden ist. Sie sollten dieses CU-Konto nicht für andere Zwecke verwenden. Sie behalten jedoch die letztendliche Kontrolle des `kmsuser`-CU-Kontos. Sie können jederzeit die Key-Handles von Schlüsseln ermitteln, über die `kmsuser` verfügt. Bei Bedarf können Sie den benutzerdefinierten Schlüsselspeicher trennen, das `kmsuser`-Passwort ändern, sich bei dem Cluster als `kmsuser` anmelden sowie die Schlüssel, über die `kmsuser` verfügt, anzeigen und verwalten.

Für Anweisungen zum Erstellen Ihres `kmsuser`-CU-Kontos vgl. Erstellen des `kmsuser`-Crypto-Benutzers.

KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher

Sie können die AWS KMS oder die AWS KMS-API zum Erstellen eines AWS KMS keys in einem AWS CloudHSM-Schlüsselspeicher verwenden. Sie verwenden die gleiche Methode wie bei jedem KMS-Schlüssel. Der einzige Unterschied besteht darin, dass Sie den AWS CloudHSM-Schlüsselspeicher identifizieren und angeben müssen, dass der AWS CloudHSM-Cluster der Ursprung des Schlüsselmaterials ist.

Wenn Sie einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher erstellen, erstellt AWS KMS den KMS-Schlüssel in AWS KMS und einen persistenten und nicht exportierbaren 256-Bit-Unterstützungsschlüssel nach Advanced Encryption Standard (AES) in dem zugehörigen Cluster. Wenn Sie den AWS KMS-Schlüssel in einer kryptografischen Operation verwenden, wird die Operation im AWS CloudHSM-Cluster mit dem clusterbasierten AES-Schlüssel durchgeführt. Obwohl AWS CloudHSM symmetrische und asymmetrische Schlüssel verschiedener Typen unterstützt, unterstützen AWS CloudHSM-Schlüsselspeicher nur symmetrische AES-Verschlüsselungsschlüssel.

Sie können die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher in der AWS KMS-Konsole anzeigen und die Konsolenoptionen verwenden, um die ID des benutzerdefinierten Schlüsselspeichers anzuzeigen. Sie können die -DescribeKeyOperation auch verwenden, um die -AWS CloudHSMSchlüsselspeicher-ID und die AWS CloudHSM Cluster-ID zu finden.

Die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher funktionieren genau wie andere KMS-Schlüssel in AWS KMS. Autorisierte Benutzer benötigen die gleichen Berechtigungen zum Verwenden und Verwalten der KMS-Schlüssel. Sie verwenden die gleichen Konsolenprozeduren und API-Operationen zum Anzeigen und Verwalten der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher. Dazu gehören das Aktivieren und Deaktivieren von KMS-Schlüsseln, das Erstellen und Verwenden von Tags und Aliasen sowie das Festlegen und Ändern von IAM- und Schlüsselrichtlinien. Sie können die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher für kryptografische Vorgänge verwenden und sie mit integrierten AWS-Services verwenden, die kundenverwaltete Schlüssel unterstützen. Sie können jedoch nicht die automatische Schlüsselrotation aktivieren oder Schlüsselmaterial in einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher importieren.

Darüber hinaus verwenden Sie den gleichen Prozess zum Planen des Löschens eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher. Nach Ablauf der Wartezeit löscht AWS KMS den KMS-Schlüssel aus KMS. Anschließend versucht es unter Verwendung aller verfügbaren

Methoden, das Schlüsselmaterial für den KMS-Schlüssel aus dem zugehörigen AWS CloudHSM-Cluster zu löschen. Möglicherweise müssen Sie jedoch das verwaiste Schlüsselmaterial manuell aus dem Cluster und seinen Backups löschen.

Steuern des Zugriffs auf Ihren AWS CloudHSM-Schlüsselspeicher

Sie verwenden IAM-Richtlinien, um den Zugriff auf Ihren AWS CloudHSM-Schlüsselspeicher und Ihr AWS CloudHSM-Cluster zu steuern. Sie können Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen verwenden, um den Zugriff auf die AWS KMS keys in Ihrem AWS CloudHSM-Schlüsselspeicher zu steuern. Sie sollten Benutzern, Gruppen und Rollen ausschließlich die Berechtigungen gewähren, die sie für die Aufgaben benötigen, die sie voraussichtlich ausführen werden.

Themen

- Autorisierung von Managern und Benutzern des AWS CloudHSM-Schlüsselspeichers
- Autorisieren von AWS KMS für die Verwaltung von AWS CloudHSM- und Amazon-EC2-Ressourcen

Autorisierung von Managern und Benutzern des AWS CloudHSM-Schlüsselspeichers

Achten Sie beim Entwerfen Ihres AWS CloudHSM-Schlüsselspeichers darauf, dass die Prinzipale, die diesen verwenden und verwalten, ausschließlich die Berechtigungen erhalten, die sie benötigen. Die folgende Liste beschreibt die Mindestberechtigungen, die Manager und Benutzer von AWS CloudHSM-Schlüsselspeichern benötigen.

- Prinzipale, die Ihre AWS CloudHSM-Schlüsselspeicher erstellen und verwalten, benötigen die folgende Berechtigung, um die API-Operationen des AWS CloudHSM-Schlüsselspeichers verwenden zu können.
  - `cloudhsm:DescribeClusters`
  - `kms:CreateCustomKeyStore`
  - `kms:ConnectCustomKeyStore`
  - `kms:DeleteCustomKeyStore`
  - `kms:DescribeCustomKeyStores`
  - `kms:DisconnectCustomKeyStore`
  - `kms:UpdateCustomKeyStore`
  - `iam:CreateServiceLinkedRole`

- Prinzipale, die den mit Ihrem AWS CloudHSM-Schlüsselspeicher verknüpften AWS CloudHSM-Cluster erstellen und verwalten, benötigen eine Berechtigung zum Erstellen und Initialisieren von AWS CloudHSM-Clustern. Dies schließt die Berechtigung zum Erstellen oder Verwenden einer Amazon Virtual Private Cloud (VPC), zum Erstellen von Subnetzen und zum Erstellen einer Amazon-EC2-Instance ein. Sie müssen möglicherweise auch HSMs erstellen und löschen und Sicherungen verwalten. Eine Liste der erforderlichen Berechtigungen finden Sie unter Identitäts- und Zugriffsverwaltung für AWS CloudHSM im AWS CloudHSM Benutzerhandbuch.
- Prinzipale, die AWS KMS keys in Ihrem AWS CloudHSM-Schlüsselspeicher erstellen und verwalten, benötigen die gleichen Berechtigungen wie Prinzipale, die KMS-Schlüssel in AWS KMS erstellen und verwalten. Die Standard-Schlüsselrichtlinie für CMKs in AWS CloudHSM-Schlüsselspeichern ist identisch mit der Standard-Schlüsselrichtlinie für CMKs in AWS KMS. Attributbasierte Zugriffssteuerung (ABAC), das Tags und Aliasse verwendet, um den Zugriff auf KMS-Schlüssel zu steuern, ist auch für KMS-Schlüssel in AWS CloudHSM-Schlüsselspeichern wirksam.
- Prinzipale, die die KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher für kryptografische Vorgänge verwenden, benötigen die Berechtigung zum Ausführen der kryptografischen Operation mit dem KMS-Schlüssel, z. B. kms:Decrypt. Sie können diese Berechtigungen in einer Schlüsselrichtlinie oder einer IAM-Richtlinie bereitstellen. Sie benötigen keine zusätzlichen Berechtigungen, um einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher verwenden zu können.

Autorisieren von AWS KMS für die Verwaltung von AWS CloudHSM- und Amazon-EC2-Ressourcen

Um Ihre AWS CloudHSM-Schlüsselspeicher zu verwalten, benötigt AWS KMS die Berechtigung zum Abrufen von Informationen zu Ihren AWS CloudHSM-Clustern. Darüber hinaus werden Berechtigungen zum Erstellen der Netzwerkinfrastruktur benötigt, die Ihren AWS CloudHSM-Schlüsselspeicher mit dessen –AWS CloudHSMCluster verbindet. Um diese Berechtigungen zu erhalten, AWS KMS erstellt die AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpfte Rolle in Ihrem AWS-Konto. Benutzer, die AWS CloudHSM-Schlüsselspeicher erstellen, müssen die Berechtigung `iam:CreateServiceLinkedRole` zum Erstellen serviceverknüpfter Rollen besitzen.

Themen

- Über die serviceverknüpfte Rolle AWS KMS
- Erstellen der serviceverknüpften Rolle
- Bearbeiten der Beschreibung der serviceverknüpften Rolle

- Löschen der serviceverknüpften Rolle

Über die serviceverknüpfte Rolle AWS KMS

Eine serviceverknüpfte Rolle ist eine IAM-Rolle, die einem AWS-Service die Berechtigung gewährt, in Ihrem Namen andere AWS-Services aufzurufen. Sie soll die Verwendung der Funktionen mehrerer integrierter AWS-Services für Sie vereinfachen, da Sie keine komplexen IAM-Richtlinien erstellen und verwalten müssen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für AWS KMS.

Für -AWS CloudHSMSchlüsselspeicher AWS KMS erstellt die AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpfte Rolle mit der AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy Richtlinie . Diese Richtlinie gewährt der Rolle die folgenden Berechtigungen:

- cloudhsm:Describe* – erkennt Änderungen in dem AWS CloudHSM Cluster, der an Ihren benutzerdefinierten Schlüsselspeicher angehängt ist.
- ec2:CreateSecurityGroup – Wird verwendet, wenn Sie einen -AWS CloudHSMSchlüsselspeicher verbinden, um die Sicherheitsgruppe zu erstellen, die den Netzwerkverkehr zwischen AWS KMS und Ihrem AWS CloudHSM Cluster ermöglicht.
- ec2:AuthorizeSecurityGroupIngress – Wird verwendet, wenn Sie einen -AWS CloudHSMSchlüsselspeicher verbinden, um den Netzwerkzugriff von AWS KMS in die VPC zu ermöglichen, die Ihren AWS CloudHSM Cluster enthält.
- ec2:CreateNetworkInterface – Wird verwendet, wenn Sie einen -AWS CloudHSMSchlüsselspeicher verbinden, um die Netzwerkschnittstelle zu erstellen, die für die Kommunikation zwischen AWS KMS und dem AWS CloudHSM Cluster verwendet wird.
- ec2:RevokeSecurityGroupEgress – Wird verwendet, wenn Sie einen -AWS CloudHSMSchlüsselspeicher verbinden, um alle ausgehenden Regeln aus der Sicherheitsgruppe zu entfernen, die AWS KMS erstellt hat.
- ec2:DeleteSecurityGroup – Wird verwendet, wenn Sie einen -AWS CloudHSMSchlüsselspeicher trennen, um Sicherheitsgruppen zu löschen, die beim Verbinden des -AWS CloudHSMSchlüsselspeichers erstellt wurden.
- ec2:DescribeSecurityGroups – Wird verwendet, um Änderungen an der Sicherheitsgruppe zu überwachen, die in der VPC AWS KMS erstellt hat, die Ihren AWS CloudHSM Cluster enthält, sodass bei Ausfällen klare Fehlermeldungen bereitstellen AWS KMS kann.

- ec2:DescribeVpcs – Wird verwendet, um Änderungen in der VPC zu überwachen, die Ihren AWS CloudHSM Cluster enthält, sodass bei Ausfällen klare Fehlermeldungen bereitstellen AWS KMS kann.
- ec2:DescribeNetworkAcls – Wird verwendet, um Änderungen an den Netzwerk-ACLs für die VPC zu überwachen, die Ihren AWS CloudHSM Cluster enthält, sodass bei Ausfällen klare Fehlermeldungen bereitstellen AWS KMS kann.
- ec2:DescribeNetworkInterfaces – Wird verwendet, um Änderungen an den Netzwerkschnittstellen zu überwachen, die in der VPC AWS KMS erstellt hat, die Ihren AWS CloudHSM Cluster enthält, sodass bei Ausfällen klare Fehlermeldungen bereitstellen AWS KMS kann.

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

Da die AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpfte Rolle nur vertraut`cks.kms.amazonaws.com`, AWS KMS kann nur diese serviceverknüpfte Rolle übernehmen. Diese Rolle ist auf die Operationen beschränkt, die AWS KMS benötigt, um Ihre AWS CloudHSM-Cluster anzuzeigen und einen AWS CloudHSM-Schlüsselspeicher mit dem zugehörigen AWS CloudHSM-Cluster zu verbinden. AWS KMS werden keine weiteren Berechtigungen erteilt. Beispielsweise ist AWS KMS nicht zum Erstellen, Verwalten oder Löschen Ihrer AWS CloudHSM-Cluster, HSMs oder Sicherungen berechtigt.

Regionen

Wie die -AWS CloudHSMSchlüsselspeicherfunktion wird die AWSServiceRoleForKeyManagementServiceCustomKeyStores Rolle in allen unterstützt, in AWS-Regionen denen AWS KMS und verfügbar AWS CloudHSM sind. Eine Liste von AWS-Regionen, die jeder Service unterstützt, finden Sie unter AWS Key Management Service-Endpunkte und -Kontingente und AWS CloudHSM-Endpunkte und -Kontingente in der Allgemeine Amazon Web Services-Referenz.

Weitere Informationen darüber, wie AWS-Services serviceverknüpften Rollen verwenden, finden Sie unter Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch.

Erstellen der serviceverknüpften Rolle

AWS KMS erstellt automatisch die AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpfte Rolle in Ihrem , AWS-Konto wenn Sie einen -AWS CloudHSMSchlüsselspeicher erstellen, sofern die Rolle noch nicht vorhanden ist. Sie können diese serviceverknüpfte Rolle nicht direkt erstellen oder direkt neu erstellen.

Bearbeiten der Beschreibung der serviceverknüpften Rolle

Sie können den Namen der Rolle oder die Richtlinienanweisungen in der serviceverknüpften Rolle AWSServiceRoleForKeyManagementServiceCustomKeyStores nicht bearbeiten. Sie können jedoch die Beschreibung der Rolle bearbeiten. Anweisungen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen der serviceverknüpften Rolle

AWS KMS löscht die AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpfte Rolle nicht aus Ihrem , AWS-Konto selbst wenn Sie alle Ihre -AWS CloudHSMSchlüsselspeicher gelöscht haben. Obwohl es derzeit kein Verfahren zum Löschen der AWSServiceRoleForKeyManagementServiceCustomKeyStores serviceverknüpften Rolle gibt, übernimmt diese Rolle AWS KMS nicht und verwendet ihre Berechtigungen nicht, es sei denn, Sie haben aktive AWS CloudHSM Schlüsselspeicher.

Verwalten eines CloudHSM-Schlüsselspeichers

Sie können einen benutzerdefinierte Schlüsselspeicher über die AWS Management Console und die AWS KMS-API verwalten. Beispielsweise können Sie benutzerdefinierte Schlüsselspeicher anzeigen, ihre Eigenschaften bearbeiten, sie von dem AWS CloudHSM-Cluster trennen, mit dem sie verknüpft sind, und löschen.

Themen

Einen AWS CloudHSM-Schlüsselspeicher erstellen

Sie können einen oder mehrere AWS CloudHSM-Schlüsselspeicher in Ihrem Konto erstellen. Jeder AWS CloudHSM-Schlüsselspeicher ist mit einem AWS CloudHSM-Cluster in derselben AWS-Konto und Region verknüpft. Vor dem Erstellen des AWS CloudHSM-Schlüsselspeichers müssen Sie einige Voraussetzungen erfüllen. Bevor Sie Ihren AWS CloudHSM-Schlüsselspeicher verwenden können, verbinden Sie ihn mit seinem AWS CloudHSM-Cluster.

> **Note**
>
> Wenn Sie versuchen, einen AWS CloudHSM-Schlüsselspeicher mit den gleichen Eigenschaftswerten wie ein vorhandener getrennter AWS CloudHSM-Schlüsselspeicher zu erstellen, erstellt AWS KMS keinen neuen AWS CloudHSM-Schlüsselspeicher und löst keine Ausnahme aus oder zeigt keinen Fehler an. Stattdessen erkennt AWS KMS das Duplikat als wahrscheinliche Folge eines Wiederholungsversuchs und gibt die ID des vorhandenen AWS CloudHSM-Schlüsselspeichers zurück.

> **Tip**
>
> Sie müssen die Verbindung für Ihren AWS CloudHSM-Schlüsselspeicher nicht sofort herstellen. Sie können die Verbindung getrennt lassen, bis Sie zur Verwendung bereit sind. Wenn Sie jedoch prüfen möchten, ob die Konfiguration ordnungsgemäß erfolgt ist, können Sie eine Verbindung herstellen, den Verbindungsstatus anzeigen und anschließend die Verbindung wieder trennen.

Themen

- Erstellen eines AWS CloudHSM-Schlüsselspeichers (Konsole)
- Einen AWS CloudHSM-Schlüsselspeicher (API) erstellen

Erfüllen der Voraussetzungen

Jeder AWS CloudHSM-Schlüsselspeicher wird durch einen AWS CloudHSM-Cluster gestützt. Zum Erstellen eines AWS CloudHSM-Schlüsselspeichers müssen Sie einen aktiven AWS CloudHSM-Cluster angeben, der noch keinem anderen Schlüsselspeicher zugeordnet ist. Sie müssen auch einen dedizierten Kryptobenutzer (CU, Crypto User) in den HSMs des Clusters erstellen, mit dem AWS KMS Schlüssel für Sie erstellen und verwalten kann.

Führen Sie vor dem Erstellen eines AWS CloudHSM-Schlüsselspeichers folgende Schritte aus:

Auswahl eines AWS CloudHSM-Clusters

Jeder AWS CloudHSM-Schlüsselspeicher ist genau einem AWS CloudHSM-Cluster zugeordnet. Wenn Sie einen AWS KMS keys in Ihrem AWS CloudHSM-Schlüsselspeicher erstellen, erstellt AWS KMS die KMS-Schlüssel-Metadaten wie z. B. eine ID und einen Amazon-Ressourcennamen (ARN) in AWS KMS. Anschließend wird das Schlüsselmaterial in den HSMs des zugehörigen Clusters erstellt. Sie können einen neuen AWS CloudHSM-Cluster erstellen oder einen vorhandenen verwenden. AWS KMS benötigt keinen exklusiven Zugriff auf den Cluster.

Der von Ihnen ausgewählte AWS CloudHSM-Cluster ist dem AWS CloudHSM-Schlüsselspeicher dauerhaft zugeordnet. Nach dem Erstellen des AWS CloudHSM-Schlüsselspeichers können Sie die Cluster-ID des zugehörigen Clusters ändern, der angegebene Cluster muss jedoch einen gemeinsamen Sicherungsverlauf mit dem ursprünglichen Cluster haben. Wenn Sie einen unabhängigen Cluster verwenden möchten, müssen Sie einen neuen AWS CloudHSM-Schlüsselspeicher erstellen.

Der AWS CloudHSM-Cluster, den Sie auswählen, muss folgende Merkmale aufweisen:

- Der Cluster muss aktiv sein.

  Sie müssen den Cluster erstellen, initialisieren, die AWS CloudHSM-Client-Software für Ihre Plattform installieren und anschließend den Cluster aktivieren. Ausführliche Anweisungen finden Sie im Abschnitt Erste SchritteAWS CloudHSM im AWS CloudHSM-Benutzerhandbuch.
- Der Cluster muss sich in demselben Konto und derselben Region wie der AWS CloudHSM-Schlüsselspeicher befinden. Sie können einen AWS CloudHSM-Schlüsselspeicher in einer Region nicht mit einem Cluster in einer anderen Region.verknüpfen. Um eine

Schlüsselinfrastruktur für mehrere Regionen zu erstellen, müssen Sie in jeder Region AWS CloudHSM-Schlüsselspeicher und Cluster erstellen.

- Der Cluster kann nicht mit einem anderen benutzerdefinierten Schlüsselspeicher desselben Kontos und derselben Region verknüpft werden. Jeder AWS CloudHSM-Schlüsselspeicher im Konto und in der Region muss mit einem anderen AWS CloudHSM-Cluster verbunden sein. Sie können keinen Cluster angeben, der bereits einem benutzerdefinierten Schlüsselspeicher zugeordnet ist, und auch keinen Cluster, der einen gemeinsamen Sicherungsverlauf mit einem zugeordneten Cluster hat. Cluster mit einem gemeinsamen Sicherungsverlauf haben dasselbe Cluster-Zertifikat. Um das Cluster-Zertifikat eines Clusters anzuzeigen, verwenden Sie die -AWS CloudHSMKonsole oder die -DescribeClustersOperation.

  Wenn Sie einen AWS CloudHSM-Cluster in einer anderen Region sichern, wird er als anderer Cluster betrachtet, und Sie können die Sicherung mit einem benutzerdefinierten Schlüsselspeicher in seiner Region verknüpfen. Die KMS-Schlüssel in den beiden benutzerdefinierten Schlüsselspeichern sind jedoch nicht interoperabel, selbst wenn sie denselben Unterstützungsschlüssel haben. AWS KMS bindet Metadaten an den Geheimtext, so dass er nur mit dem KMS-Schlüssel entschlüsselt werden kann, der ihn verschlüsselt hat.

- Der Cluster muss mit privaten Subnetzen in mindestens zwei Availability Zones in der Region konfiguriert werden. Da AWS CloudHSM nicht in allen Availability Zones unterstützt wird, empfiehlt es sich, private Subnetze in allen Availability Zones in der Region zu erstellen. Sie können die Subnetze für einen vorhandenen Cluster nicht neu konfigurieren, haben jedoch die Möglichkeit, einen Cluster von einer Sicherung zu erstellen, mit anderen Subnetzen in der Cluster-Konfiguration.

  > **Important**
  >
  > Löschen Sie nach dem Erstellen des AWS CloudHSM-Schlüsselspeichers keine der für den AWS CloudHSM-Cluster konfigurierten privaten Subnetze. Wenn AWS KMS nicht alle Subnetze in der Clusterkonfiguration finden kann, schlagen Versuche, eine Verbindung mit dem benutzerdefinierten Schlüsselspeicher herzustellen, mit einem `SUBNET_NOT_FOUND`-Verbindungsfehlerstatus fehl. Details hierzu finden Sie unter Beheben eines Verbindungsfehlers.

- Die Sicherheitsgruppe für den Cluster (`cloudhsm-cluster-<cluster-id>-sg`) muss Regeln für ein- und ausgehenden Datenverkehr enthalten, die TCP-Datenverkehr über die Ports 2223-2225 erlauben. Die Angabe für Source (Quelle) in den eingehenden Regeln und für Destination (Ziel) in den ausgehenden Regeln muss mit der Sicherheitsgruppen-

ID übereinstimmen. Diese Regeln werden standardmäßig festgelegt, wenn Sie den Cluster erstellen. Sie dürfen nicht gelöscht oder geändert werden.

- DerCluster muss über mindestens zwei aktive HSMs in verschiedenen Availability Zones verfügen. Um die Anzahl der HSMs zu überprüfen, verwenden Sie die -Konsole oder die -DescribeClustersOperation. AWS CloudHSM Falls erforderlich, können Sie ein HSM hinzufügen.

Finden des Trust Anchor-Zertifikats

Wenn Sie einen benutzerdefinierten Schlüsselspeicher erstellen, müssen Sie das Trust Anchor-Zertifikat für den AWS CloudHSM-Cluster in AWS KMS hochladen. AWS KMS benötigt das Trust Anchor-Zertifikat, um den AWS CloudHSM-Schlüsselspeicher mit dem zugehörigen AWS CloudHSM-Cluster zu verbinden.

Jeder aktive AWS CloudHSM-Cluster verfügt über ein Trust Anchor-Zertifikat. Wenn Sie den Cluster initialisieren, generieren Sie dieses Zertifikat, speichern es in der Datei `customerCA.crt` und kopieren es auf Hosts, die eine Verbindung zu dem Cluster herstellen.

Erstellen des **`kmsuser`**-Kryptobenutzers für AWS KMS

Zur Verwaltung Ihres AWS CloudHSM-Schlüsselspeichers meldet sich AWS KMS beim Konto des `kmsuser`-Kryptobenutzers (CU, Crypto User) im ausgewählten Cluster an. Vor dem Erstellen Ihres AWS CloudHSM Schlüsselspeichers müssen Sie den `kmsuser`-CU erstellen. Wenn Sie dann den AWS CloudHSM-Schlüsselspeicher erstellen, geben Sie AWS KMS das Passwort für `kmsuser`. Wenn Sie den AWS CloudHSM-Schlüsselspeicher mit dem zugehörigen AWS CloudHSM-Cluster verbinden, meldet sich AWS KMS im Cluster als der `kmsuser` an und rotiert das `kmsuser`-Passwort.

> **Important**
>
> Geben Sie nicht die Option `2FA` beim Erstellen des `kmsuser`-CU an. Wenn Sie dies tun, kann sich AWS KMS nicht anmelden und Ihr AWS CloudHSM-Schlüsselspeicher kann nicht mit diesem AWS CloudHSM-Cluster verbunden werden. Wenn Sie 2FA angeben, können Sie dies nicht rückgängig machen. Sie müssen dann den CU löschen und neu erstellen.

Gehen Sie wie folgt vor, um den `kmsuser`-CU zu erstellen.

1. Starten Sie cloudhsm_mgmt_util wie im Abschnitt Erste Schritte mit CloudHSM Management Utility (CMU) des AWS CloudHSM-Benutzerhandbuchs beschrieben.
2. Verwenden Sie den Befehl createUser in cloudhsm_mgmt_util zum Erstellen eines CU mit dem Namen `kmsuser`. Das Passwort muss 7 – 32 alphanumerische Zeichen umfassen. Bei der Angabe wird zwischen Groß- und Kleinschreibung unterschieden, Sonderzeichen sind nicht zulässig.

   Mit dem folgenden Befehl beispielsweise wird ein `kmsuser`-CU mit dem Passwort `kmsPswd` erstellt.

   ```
   aws-cloudhsm> createUser CU kmsuser kmsPswd
   ```

Erstellen eines AWS CloudHSM-Schlüsselspeichers (Konsole)

Wenn Sie einen AWS CloudHSM-Schlüsselspeicher in der AWS Management Console erstellen, können Sie die erforderlichen Komponenten im Rahmen Ihres Workflow hinzufügen und erstellen. Der Prozess ist jedoch schneller, wenn Sie die Voraussetzungen schon vorab erfüllt haben.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.
4. Wählen Sie Einen Schlüsselspeicher erstellen aus.
5. Geben Sie einen Anzeigenamen für den benutzerdefinierten Schlüsselspeicher ein. Der Name muss unter allen beutzerdefinierten Schlüsselspeichern in Ihrem Konto eindeutig sein.

   > **Important**
   >
   > Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

6. Wählen Sie einen AWS CloudHSM-Cluster für den AWS CloudHSM-Schlüsselspeicher aus. Sie können auch einen neuen AWS CloudHSM-Cluster erstellen. Wählen Sie in diesem Fall den Link Create an AWS CloudHSM-Cluster (HSM-Cluster erstellen) aus.

   Im Menü werden die AWS CloudHSM-Cluster in Ihrem Konto und Ihrer Region angezeigt, die noch keinem AWS CloudHSM-Schlüsselspeicher zugeordnet sind. Der Cluster muss die Anforderungen für die Zuordnung zu einem benutzerdefinierten Schlüsselspeicher erfüllen.
7. Wählen Sie Choose file (Datei auswählen) aus und laden Sie das Trust Anchor-Zertifikat für den von Ihnen ausgewählten AWS CloudHSM-Cluster hoch. Dies ist die Datei `customerCA.crt`, die Sie bei der Initialisierung des Clusters erstellt haben.
8. Geben Sie das Passwort des `kmsuser`-Kryptobenutzers (CU) ein, den Sie in dem ausgewählten Cluster erstellt haben.
9. Wählen Sie Erstellen.

Bei erfolgreicher Ausführung wird der neue AWS CloudHSM-Schlüsselspeicher in der Liste der AWS CloudHSM-Schlüsselspeicher in dem Konto und der Region aufgeführt. Bei nicht erfolgreicher Ausführung wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung für einen Custom Key Store.

Wenn Sie versuchen, einen AWS CloudHSM-Schlüsselspeicher mit den gleichen Eigenschaftswerten wie ein vorhandener getrennter AWS CloudHSM-Schlüsselspeicher zu erstellen, erstellt AWS KMS keinen neuen AWS CloudHSM-Schlüsselspeicher und löst keine Ausnahme aus oder zeigt keinen Fehler an. Stattdessen erkennt AWS KMS das Duplikat als wahrscheinliche Folge eines Wiederholungsversuchs und gibt die ID des vorhandenen AWS CloudHSM-Schlüsselspeichers zurück.

Nächster Schritt: Neue AWS CloudHSM-Schlüsselspeicher werden nicht automatisch verbunden. Bevor Sie im AWS CloudHSM-Schlüsselspeicher AWS KMS keys erstellen können, verbinden Sie den benutzerdefinierten Schlüsselspeicher zu seinem zugewiesenen –AWS CloudHSM-Cluster.

Einen AWS CloudHSM-Schlüsselspeicher (API) erstellen

Sie können die -CreateCustomKeyStoreOperation verwenden, um einen neuen AWS CloudHSM Schlüsselspeicher zu erstellen, der einem -AWS CloudHSMCluster im Konto und in der Region zugeordnet ist. Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Für die Produktion `CreateCustomKeyStore` sind folgende Parameterwerte erforderlich.

- CustomKeyStoreName – Ein Anzeigename für den benutzerdefinierten Schlüsselspeicher, der im Konto eindeutig ist.

  > **Important**
  >
  > Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

- CloudHsmClusterId – Die Cluster-ID eines -AWS CloudHSMClusters, der die Anforderungen für einen -Schlüsselspeicher erfüllt. AWS CloudHSM
- KeyStorePassword – Das Passwort des `kmsuser` CU-Kontos im angegebenen Cluster.
- TrustAnchorCertificate – Der Inhalt der -`customerCA.crt`Datei, die Sie beim Initialisieren des Clusters erstellt haben.

Im folgenden Beispiel wird eine fiktive Cluster-ID verwendet. Ersetzen Sie diese vor Ausführung des Befehls durch eine gültige Cluster-ID.

```
$ 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>
```

Wenn Sie die AWS CLI verwenden, können Sie die Trust Anchor-Zertifikatsdatei anstatt ihres Inhalts angeben. Im folgenden Beispiel befindet sich die Datei `customerCA.crt` im Stammverzeichnis.

```
$ 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
```

Bei einer erfolgreichen Produktion gibt `CreateCustomKeyStore` die ID des benutzerdefinierten Schlüsselspeichers zurück, wie in der folgenden Beispielantwort dargestellt.

```
{
```

```
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Wenn die Produktion fehlschlägt, korrigieren Sie den in der Ausnahme angegebenen Fehler und versuchen Sie es erneut. Weitere Informationen finden Sie unter Fehlerbehebung für einen Custom Key Store.

Wenn Sie versuchen, einen AWS CloudHSM-Schlüsselspeicher mit den gleichen Eigenschaftswerten wie ein vorhandener getrennter AWS CloudHSM-Schlüsselspeicher zu erstellen, erstellt AWS KMS keinen neuen AWS CloudHSM-Schlüsselspeicher und löst keine Ausnahme aus oder zeigt keinen Fehler an. Stattdessen erkennt AWS KMS das Duplikat als wahrscheinliche Folge eines Wiederholungsversuchs und gibt die ID des vorhandenen AWS CloudHSM-Schlüsselspeichers zurück.

Nächster Schritt: Um den AWS CloudHSM-Schlüsselspeicher verwenden zu können, verbinden Sie diesen nun mit seinem AWS CloudHSM-Cluster.

Anzeigen eines AWS CloudHSM-Schlüsselspeichers

Sie können die -AWS CloudHSMSchlüsselspeicher in jedem Konto und jeder Region mithilfe der -AWS KMSKonsole oder der -DescribeCustomKeyStoresOperation anzeigen.

Weitere Informationen finden Sie auch unter:

- Anzeigen eines externen Schlüsselspeichers
- Anzeigen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher
- AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail

Themen

- Anzeigen eines AWS CloudHSM-Schlüsselspeichers (Konsole)
- Anzeigen eines AWS CloudHSM-Schlüsselspeichers (API)

Anzeigen eines AWS CloudHSM-Schlüsselspeichers (Konsole)

Wenn Sie die AWS CloudHSM-Schlüsselspeicher in der AWS Management Console anzeigen, werden die folgenden Informationen angezeigt:

- Name und ID des benutzerdefinierten Schlüsselspeichers

- Die ID des zugeordneten AWS CloudHSM-Clusters
- Die Anzahl der HSM im Cluster
- Der aktuelle Verbindungsstatus

Der Verbindungsstatus (Status) Disconnected (Verbindung getrennt) gibt an, dass der benutzerdefinierte Schlüsselspeicher neu ist und noch nicht verbunden war oder dass absichtlich die Verbindung zu seinem AWS CloudHSM-Cluster getrennt wurde. Wenn Ihre Versuche, einen KMS-Schlüssel in einem verbundenen benutzerdefinierten Schlüsselspeicher zu verwenden, jedoch fehlschlagen, kann dies auf ein Problem mit dem benutzerdefinierten Schlüsselspeicher oder dem AWS CloudHSM-Cluster hindeuten. Weitere Informationen dazu finden Sie unter Beheben eines fehlerhaften KMS-Schlüssels.

Gehen Sie wie folgt vor, um die AWS CloudHSM-Schlüsselspeicher in einem bestimmten Konto und einer bestimmten Region anzuzeigen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.

Klicken Sie auf das Zahnradsymbol unter der Schaltfläche Create key store (Schlüsselspeicher erstellen), um die Anzeige anzupassen.

Anzeigen eines AWS CloudHSM-Schlüsselspeichers (API)

Um Ihre -AWS CloudHSMSchlüsselspeicher anzuzeigen, verwenden Sie die -DescribeCustomKeyStoresOperation. Standardmäßig gibt diese Operation alle benutzerdefinierten Schlüsselspeicher im Konto und in der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die Ausgabe auf einen bestimmten benutzerdefinierten Schlüsselspeicher zu begrenzen. Bei AWS CloudHSM-Schlüsselspeichern besteht die Ausgabe aus der ID, dem Namen und dem Typ des benutzerdefinierten Schlüsselspeichers, der ID der zugehörigen AWS CloudHSM-Clusters und dem Verbindungsstatus. Wenn der Verbindungsstatus auf einen Fehler hinweist, enthält die Ausgabe außerdem einen Fehlercode, der die Ursache angibt.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Beispielsweise gibt der folgende Befehl alle benutzerdefinierten Schlüsselspeicher im Konto und in der Region zurück. Sie können die Parameter `Marker` und `Limit` verwenden, um durch die benutzerdefinierten Schlüsselspeicher in der Ausgabe zu blättern.

```
$ aws kms describe-custom-key-stores
```

Der folgende Beispielbefehl verwendet den Parameter `CustomKeyStoreName`, um nur den benutzerdefinierten Schlüsselspeicher mit dem Anzeigenamen `ExampleCloudHSMKeyStore` abzurufen. In jedem Befehl können Sie entweder den Parameter `CustomKeyStoreName` oder `CustomKeyStoreId` (aber nicht beide) verwenden.

Die folgende Beispielausgabe stellt einen AWS CloudHSM-Schlüsselspeicher dar, der mit seinem AWS CloudHSM-Cluster verbunden ist.

> **Note**
>
> Das Feld `CustomKeyStoreType` wurde der Antwort `DescribeCustomKeyStores` hinzugefügt, um AWS CloudHSM-Schlüsselspeicher von externen Schlüsselspeichern zu unterscheiden.

```
$ 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>"
      }
   ]
}
```

Der `ConnectionState` `Disconnected` gibt an, dass ein benutzerdefinierter Schlüsselspeicher noch nicht verbunden war oder dass absichtlich die Verbindung zum AWS CloudHSM-Cluster getrennt wurde. Wenn Versuche, einen KMS-Schlüssel in einem verbundenen AWS CloudHSM-Schlüsselspeicher zu verwenden, jedoch fehlschlagen, kann dies auf ein Problem mit dem AWS CloudHSM-Schlüsselspeicher oder seinem AWS CloudHSM-Cluster hindeuten. Weitere Informationen dazu finden Sie unter Beheben eines fehlerhaften KMS-Schlüssels.

Wenn der `ConnectionState` des benutzerdefinierten Schlüsselspeichers `FAILED` lautet, enthält die `DescribeCustomKeyStores`-Antwort ein `ConnectionErrorCode`-Element, das den Grund für den Fehler angibt.

In der folgenden Ausgabe deutet der Wert `INVALID_CREDENTIALS` beispielsweise darauf hin, dass die Verbindung des benutzerdefinierten Schlüsselspeichers fehlgeschlagen ist, da das `kmsuser`-Passwort ungültig ist. Weitere Informationen zu diesem und anderen Verbindungsfehlern finden Sie unter Fehlerbehebung für einen Custom Key Store.

```
$ 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>"
      }
   ]
}
```

Einstellungen des AWS CloudHSM-Schlüsselspeichers bearbeiten

Sie können die Einstellungen eines vorhandenen AWS CloudHSM-Schlüsselspeichers ändern. Der benutzerdefinierte Schlüsselspeicher muss von dessen AWS CloudHSM-Cluster getrennt sein.

So bearbeiten Sie die Einstellungen desAWS CloudHSM-Schlüsselspeichers:

1. Trennen Sie den benutzerdefinierten Schlüsselspeicher von dessen AWS CloudHSM-Cluster. Während der benutzerdefinierte Schlüsselspeicher getrennt ist, können Sie im benutzerdefinierten

Schlüsselspeicher keine AWS KMS keys (KMS-Schlüssel) erstellen und die enthaltenen KMS-Schlüssel nicht für kryptografische Operationen verwenden.

2. Bearbeiten Sie eine oder mehrere der Einstellungen für den AWS CloudHSM-Schlüsselspeicher.
3. Verbinden Sie den benutzerdefinierten Schlüsselspeicher wieder mit dessen AWS CloudHSM-Cluster.

Sie können in einem benutzerdefinierten Schlüsselspeicher die folgenden Einstellungen ändern:

Anzeigename des benutzerdefinierten Schlüsselspeichers.

Geben Sie einen neuen Anzeigenamen ein. Der neue Name muss unter allen benutzerdefinierten Schlüsselspeichern in Ihrem AWS-Konto eindeutig sein.

> **Important**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

Cluster-ID des verknüpften AWS CloudHSM-Clusters.

Bearbeiten Sie diesen Wert, um das ursprüngliche Cluster durch ein verwandtes AWS CloudHSM-Cluster zu ersetzen. Sie können dieses Feature verwenden, um einen benutzerdefinierten Schlüsselspeicher zu reparieren, wenn dessen AWS CloudHSM-Cluster beschädigt oder gelöscht wurde.

Geben Sie ein AWS CloudHSM-Cluster ein, das denselben Sicherungsverlauf wie das ursprüngliche Cluster hat und die Anforderungen in Bezug auf Verknüpfungen mit benutzerdefinierten Schlüsselspeichern erfüllt, einschließlich des Vorhandenseins von zwei aktiven HSMs in verschiedenen Availability Zones. Cluster mit einem gemeinsamen Sicherungsverlauf haben dasselbe Cluster-Zertifikat. Um das Cluster-Zertifikat eines Clusters anzuzeigen, verwenden Sie die -DescribeClustersOperation. Sie können das Bearbeitungs-Feature nicht verwenden, um den benutzerdefinierten Schlüsselspeicher mit einem nicht verwandten AWS CloudHSM-Cluster zu verknüpfen.

Das aktuelle Passwort des **`kmsuser`** Kryptobenutzers (Crypto User, CU).

Teilt AWS KMS das aktuelle Passwort des `kmsuser`-CU im AWS CloudHSM-Cluster mit. Durch diese Aktion wird das Passwort des `kmsuser`-CU im AWS CloudHSM-Cluster nicht geändert.

Wenn Sie das Passwort des `kmsuser`-CU im AWS CloudHSM-Cluster ändern, verwenden Sie dieses Feature, um AWS KMS das neue `kmsuser`-Passwort mitzuteilen. Andernfalls kann sich AWS KMS nicht am Cluster anmelden und jeder Versuch, den benutzerdefinierten Schlüsselspeicher mit dem Cluster zu verbinden, schlägt fehl.

Themen

- Bearbeiten eines AWS CloudHSM-Schlüsselspeichers (Konsole)
- Einen AWS CloudHSM-Schlüsselspeicher (API) bearbeiten

Bearbeiten eines AWS CloudHSM-Schlüsselspeichers (Konsole)

Wenn Sie einen AWS CloudHSM-Schlüssel bearbeiten, können Sie die konfigurierbaren Werte ändern.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.
4. Wählen Sie die Zeile des AWS CloudHSM-Schlüsselspeichers aus, den Sie bearbeiten möchten.

   Wenn der Wert in der Spalte Verbindungsstatus nicht Getrennt lautet, müssen Sie den benutzerdefinierten Schlüsselspeicher trennen, um ihn bearbeiten zu können. (Wählen Sie im Menü Key store actions(Key Store-Aktionen) die Option Disconnect (Trennen) aus.)

   Solange die Verbindung eines AWS CloudHSM-Schlüsselspeichers getrennt ist, können Sie den AWS CloudHSM-Schlüsselspeicher und seine (KMS-Schlüssel) zwar verwalten, jedoch keine KMS-Schlüssel in dem AWS CloudHSM-Schlüsselspeicher erstellen oder verwenden.
5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Edit (Bearbeiten) aus.
6. Führen Sie eine oder mehrere der folgenden Aktionen aus.

   - Geben Sie einen neuen Anzeigenamen für den benutzerdefinierten Schlüsselspeicher ein.
   - Geben Sie die Cluster-ID eines verwandten AWS CloudHSM-Clusters ein.

- Geben Sie das aktuelle Passwort des `kmsuser` Kryptobenutzers im verknüpften AWS CloudHSM-Cluster ein.

7. Wählen Sie Speichern.

   Wenn der Vorgang erfolgreich ist, wird Ihnen eine Meldung mit einer Beschreibung der von Ihnen bearbeiteten Einstellungen angezeigt. Wenn der Vorgang nicht erfolgreich ist, wird Ihnen eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung für einen Custom Key Store.

8. Stellen Sie die Verbindung des benutzerdefinierten Schlüsselspeichers wieder her.

   Um den AWS CloudHSM-Schlüsselspeicher verwenden zu können, müssen Sie ihn nach dem Bearbeiten erneut verbinden. Sie können den AWS CloudHSM-Schlüsselspeicher getrennt lassen. Während er getrennt ist können Sie jedoch im AWS CloudHSM-Schlüsselspeicher keine KMS-Schlüssel erstellen oder die KMS-Schlüssel im AWS CloudHSM-Schlüsselspeicher für kryptografische Vorgänge verwenden.

Einen AWS CloudHSM-Schlüsselspeicher (API) bearbeiten

Um die Eigenschaften eines -AWS CloudHSMSchlüsselspeichers zu ändern, verwenden Sie die -UpdateCustomKeyStoreOperation. Sie können im selben Befehl mehrere Eigenschaften eines benutzerdefinierten Schlüsselspeichers ändern. Wenn die Operation erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück. Um zu überprüfen, ob die Änderungen wirksam sind, verwenden Sie die -DescribeCustomKeyStoresOperation.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Verwenden Sie zunächst DisconnectCustomKeyStore, um den benutzerdefinierten Schlüsselspeicher von seinem Cluster zu trennen. AWS CloudHSM Ersetzen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers, cks-1234567890abcdef0, durch eine tatsächliche ID.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Im ersten Beispiel wird verwendetUpdateCustomKeyStore, um den Anzeigenamen des -AWS CloudHSMSchlüsselspeichers in zu ändern`DevelopmentKeys`. Der Befehl verwendet den

Parameter `CustomKeyStoreId`, um den AWS CloudHSM-Schlüsselspeicher zu identifizieren, und den Parameter `CustomKeyStoreName`, um den neuen Namen für den benutzerdefinierten Schlüsselspeicher anzugeben.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DevelopmentKeys
```

Im folgenden Beispiel wird der mit einem AWS CloudHSM-Schlüsselspeicher verknüpfte Cluster in eine andere Sicherung desselben Clusters geändert. Der Befehl verwendet den Parameter `CustomKeyStoreId`, um den AWS CloudHSM-Schlüsselspeicher zu identifizieren, und den Parameter `CloudHsmClusterId`, um die neue Cluster-ID anzugeben.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Im folgenden Beispiel wird AWS KMS mitgeteilt, dass das aktuelle `kmsuser`-Passwort `ExamplePassword` lautet. Der Befehl verwendet den Parameter `CustomKeyStoreId`, um den AWS CloudHSM-Schlüsselspeicher zu identifizieren, und den Parameter `KeyStorePassword`, um das aktuelle Passwort anzugeben.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

Der letzte Befehl stellt die Verbindung des AWS CloudHSM-Schlüsselspeichers mit seinem AWS CloudHSM-Cluster wieder her. Sie können den benutzerdefinierten Schlüsselspeicher im getrennten Status belassen. Sie müssen die Verbindung jedoch wiederherstellen, bevor Sie neue KMS-Schlüssel erstellen oder vorhandene KMS-Schlüssel für kryptografische Operationen verwenden können. Ersetzen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers durch eine tatsächliche ID.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Herstellen und Trennen der Verbindung eines AWS CloudHSM-Schlüsselspeichers

Neue AWS CloudHSM-Schlüsselspeicher sind nicht verbunden. Bevor Sie AWS KMS keys in Ihrem AWS CloudHSM-Schlüsselspeicher erstellen und verwenden können, müssen Sie diesen mit dem zugewiesenen AWS CloudHSM-Cluster verbinden. Es ist jederzeit möglich, die Verbindung Ihres AWS CloudHSM-Schlüsselspeichers herzustellen und zu trennen und den Verbindungsstatus anzuzeigen.

Sie müssen Ihren AWS CloudHSM-Schlüsselspeicher nicht verbinden. Sie können die Verbindung eines AWS CloudHSM-Schlüsselspeichers auf unbestimmte Zeit getrennt lassen und die Verbindung nur herstellen, wenn Sie den Schlüsselspeicher verwenden müssen. Möglicherweise möchten Sie die Verbindung jedoch von Zeit zu Zeit testen, um zu prüfen, ob die Einstellungen korrekt sind und eine Verbindungsherstellung möglich ist.

> **Note**
>
> AWS CloudHSM-Schlüsselspeicher haben nur dann einen `DISCONNECTED` Verbindungsstatus, wenn der Schlüsselspeicher noch nie verbunden wurde oder Sie ihn explizit trennen. Wenn Ihr AWS CloudHSM-Schlüsselspeicher den Status `CONNECTED` hat und Sie dennoch Probleme mit der Verwendung haben, stellen Sie sicher, dass das zugewiesene AWS CloudHSM-Cluster aktiv ist und mindestens eine aktive HSMs enthält. Hilfestellung bei fehlgeschlagenen Verbindungen finden Sie unter the section called “Fehlerbehebung für einen Custom Key Store”.

Themen

- Einen AWS CloudHSM-Schlüsselspeicher verbinden
- Die Verbindung mit einem AWS CloudHSM-Schlüsselspeicher trennen
- Herstellen einer Verbindung mit einem AWS CloudHSM-Schlüsselspeicher (Konsole)
- Herstellen einer Verbindung für einen benutzerdefinierten Schlüsselspeicher (API)
- Trennen der Verbindung für einen AWS CloudHSM-Schlüsselspeicher (Konsole)
- Trennen der Verbindung für einen AWS CloudHSM-Schlüsselspeicher (API)

Einen AWS CloudHSM-Schlüsselspeicher verbinden

Wenn Sie für einen AWS CloudHSM-Schlüsselspeicher eine Verbindung herstellen, findet AWS KMS das zugehörige AWS CloudHSM-Cluster, stellt eine Verbindung zu diesem her, meldet sich beim AWS CloudHSM-Client als `kmsuser`-Krypouser (Crypto User, CU) an und rotiert dann das `kmsuser`-Passwort. AWS KMS bleibt beim AWS CloudHSM-Client angemeldet, solange der AWS CloudHSM-Schlüsselspeicher verbunden ist.

Zum Herstellen der Verbindung erstellt AWS KMS erstellt eine Sicherheitsgruppe mit dem Namen `kms-<custom key store ID>` in der Virtual Private Cloud (VPC) des Clusters. Die Sicherheitsgruppe verfügt über eine einzige Regel, die eingehenden Datenverkehr von der Cluster-

Sicherheitsgruppe zulässt. AWS KMS erstellt außerdem eine Elastic Network-Schnittstelle (ENI, Elastic Network Interface) in jeder Availability Zone des privaten Subnetzes für den Cluster. AWS KMS fügt die ENIs der `kms-<cluster ID>`-Sicherheitsgruppe und der Sicherheitsgruppe für den Cluster hinzu. Die Beschreibung der einzelnen ENIs lautet `KMS managed ENI for cluster <cluster-ID>`.

Der Verbindungsvorgang kann einige Zeit – bis zu 20 Minuten – in Anspruch nehmen.

Bevor Sie eine Verbindung für den AWS CloudHSM-Schlüsselspeicher herstellen, prüfen Sie, ob dieser die Anforderungen erfüllt.

- Der zugehörige AWS CloudHSM-Cluster muss mindestens ein aktives HSM enthalten. Um die Anzahl der HSMs im Cluster zu ermitteln, zeigen Sie den Cluster in der -AWS CloudHSMKonsole an oder verwenden Sie die -DescribeClustersOperation. Falls erforderlich, können Sie ein HSM hinzufügen.
- Der Cluster muss über ein `kmsuser`-Kryptobenutzer (Crypto User, CU)-Konto verfügen, aber dieser CU darf nicht beim Cluster angemeldet sein, wenn Sie die Verbindung zum AWS CloudHSM-Schlüsselspeicher herstellen. Hilfe zum Abmelden finden Sie unter Abmelden und erneutes Verbinden.
- Der Verbindungsstatus des AWS CloudHSM-Schlüsselspeichers kann nicht `DISCONNECTING` oder `FAILED` lauten. Um den Verbindungsstatus anzuzeigen, verwenden Sie die -AWS KMSKonsole oder die -DescribeCustomKeyStoresAntwort. Wenn der Verbindungsstatus `FAILED` lautet, trennen Sie den benutzerdefinierten Schlüsselspeicher, lösen Sie das Problem und stellen Sie anschließend die Verbindung her.

Hilfestellung bei fehlgeschlagenen Verbindungen finden Sie unter Beheben eines Verbindungsfehlers.

Wenn Ihr AWS CloudHSM-Schlüsselspeicher verbunden ist, können Sie KMS-Schlüssel darin erstellen und vorhandene KMS-Schlüssel in kryptografischen Produktionen verwenden.

Die Verbindung mit einem AWS CloudHSM-Schlüsselspeicher trennen

Wenn Sie die Verbindung eines AWS CloudHSM-Schlüsselspeichers trennen, meldet sich AWS KMS beim AWS CloudHSM-Client ab, trennt die Verbindung zu dem zugehörigen AWS CloudHSM-Cluster und entfernt die zur Unterstützung der Verbindung erstellte Netzwerkinfrastruktur.

Solange die Verbindung eines AWS CloudHSM-Schlüsselspeichers getrennt ist, können Sie den AWS CloudHSM-Schlüsselspeicher und seine (KMS-Schlüssel) zwar verwalten, jedoch

keine KMS-Schlüssel in dem AWS CloudHSM-Schlüsselspeicher erstellen oder verwenden. Der Verbindungsstatus des Schlüsselspeichers lautet `DISCONNECTED` und der Schlüsselstatus der KMS-Schlüssel in dem benutzerdefinierten Schlüsselspeicher `Unavailable`, sofern er nicht `PendingDeletion` lautet. Sie können die Verbindung des AWS CloudHSM-Schlüsselspeichers jederzeit wiederherstellen.

Wenn Sie die Verbindung zu einem benutzerdefinierten Schlüsselspeicher trennen, werden die KMS-Schlüssel im Schlüsselspeicher sofort unbrauchbar (je nach letztendlicher Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel geschützten Datenschlüsseln verschlüsselt wurden, sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, von denen viele Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

> **Note**
>
> Sämtliche Versuche, KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen oder vorhandene KMS-Schlüssel in kryptografischen Produktionen zu nutzen, schlagen fehl, während der benutzerdefinierte Schlüsselspeicher getrennt ist. Diese Aktion kann verhindern, dass Benutzer vertrauliche Daten speichern und darauf zugreifen.

Um die Auswirkung einer Trennung der Verbindung Ihres benutzerdefinierten Schlüsselspeichers besser beurteilen zu können, ermitteln Sie die KMS-Schlüssel im benutzerdefinierten Schlüsselspeicher und ihre bisherige Verwendung.

Die Verbindung des AWS CloudHSM-Schlüsselspeichers könnte beispielsweise aus folgenden Gründen getrennt werden:

- Zum Rotieren des **`kmsuser`**-Passworts. AWS KMS ändert das `kmsuser`-Passwort jedes Mal, wenn eine Verbindung zu dem AWS CloudHSM-Cluster hergestellt wird. Um eine Passwort-Rotation zu erzwingen, trennen Sie einfach die Verbindung und stellen Sie sie wieder her.
- Zur Prüfung des Schlüsselmaterials für die KMS-Schlüssel in dem AWS CloudHSM-Cluster. Wenn Sie die Verbindung des benutzerdefinierten Schlüsselspeichers trennen, meldet sich AWS KMS vom `kmsuser`-Kryptobenutzer-Konto im AWS CloudHSM-Client ab. So können Sie sich als `kmsuser`-CU beim Cluster anmelden und das Schlüsselmaterial für den KMS-Schlüssel prüfen und verwalten.

- Zum sofortigen Deaktivieren aller KMS-Schlüssel im AWS CloudHSM Schlüsselspeicher. Sie können KMS-Schlüssel in einem -Schlüsselspeicher deaktivieren und wieder aktivieren, indem Sie die DisableKey Operation AWS Management Console oder verwenden. AWS CloudHSM Diese Produktionen werden schnell ausgeführt, beziehen sich jedoch immer nur auf jeweils einen KMS-Schlüssel. Beim Trennen der Verbindung wird der AWS CloudHSM Schlüsselstatus aller KMS-Schlüssel in dem AWS CloudHSM Schlüssel umgehend in `Unavailable` geändert, wodurch sie nicht mehr in kryptografischen Produktionen verwendet werden können.
- Zur Behebung eines fehlgeschlagenen Verbindungsversuchs. Wenn ein Versuch, eine Verbindung für einen AWS CloudHSM-Schlüsselspeicher herzustellen, fehlschlägt (Verbindungsstatus des benutzerdefinierten Schlüsselspeichers: `FAILED`), müssen Sie die Verbindung des AWS CloudHSM-Schlüsselspeichers trennen, bevor Sie erneut versuchen, eine Verbindung herzustellen.

Herstellen einer Verbindung mit einem AWS CloudHSM-Schlüsselspeicher (Konsole)

Um in der AWS Management Console eine Verbindung für einen AWS CloudHSM-Schlüsselspeicher herzustellen, wählen Sie den AWS CloudHSM-Schlüsselspeicher zunächst auf der Seite Custom key stores (Benutzerdefinierte Schlüsselspeicher) aus. Der Verbindungsvorgang kann bis zu 20 Minuten dauern.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.
4. Wählen Sie die Zeile des AWS CloudHSM-Schlüsselspeichers aus, mit dem Sie eine Verbindung herstellen möchten.

   Wenn der Status des AWS CloudHSM-Schlüsselspeichers Fehlgeschlagen lautet, müssen Sie den benutzerdefinierten Schlüsselspeicher trennen, bevor Sie die Verbindung herstellen.
5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Connect (Verbinden) aus.

AWS KMS beginnt, die Verbindung für Ihren benutzerdefinierten Schlüsselspeicher herzustellen. Der Service findet den zugehörigen AWS CloudHSM-Cluster, schafft die erforderlichen Netzwerkinfrastruktur, stellt eine Verbindung her, meldet sich beim AWS CloudHSM-Cluster als

`kmsuser`-CU an und rotiert das `kmsuser`-Passwort. Wenn die Produktion abgeschlossen ist, ändert sich der Verbindungsstatus in Verbunden.

Wenn die Produktion fehlschlägt, wird eine Fehlermeldung mit einer Beschreibung der Fehlerursache angezeigt. Bevor Sie versuchen, erneut eine Verbindung herzustellen, zeigen Sie den Verbindungsstatus Ihres AWS CloudHSM-Schlüsselspeichers an. Wenn dieser Fehlgeschlagen lautet, müssen Sie die Verbindung des benutzerdefinierten Schlüsselspeichers trennen, bevor Sie erneut eine Verbindung herstellen. Wenn Sie Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung für einen Custom Key Store.

Danach: the section called “Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher”.

Herstellen einer Verbindung für einen benutzerdefinierten Schlüsselspeicher (API)

Um eine Verbindung zu einem getrennten AWS CloudHSM Schlüsselspeicher herzustellen, verwenden Sie die -ConnectCustomKeyStoreOperation. Der zugeordnete AWS CloudHSM-Cluster muss mindestens ein aktives HSM enthalten und der Verbindungsstatus darf nicht `FAILED` lauten.

Der Verbindungsvorgang nimmt einige Zeit – bis zu 20 Minuten – in Anspruch. Sofern sie nicht schnell fehlschlägt, gibt die Produktion eine HTTP-Antwort 200 und ein JSON-Objekt ohne Eigenschaften zurück. Diese erste Antwort gibt jedoch nicht an, dass die Verbindung erfolgreich war. Informationen zum Ermitteln des Verbindungsstatus des benutzerdefinierten Schlüsselspeichers finden Sie in der DescribeCustomKeyStores Antwort.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Identifizieren Sie den AWS CloudHSM-Schlüsselspeicher anhand der ID des benutzerdefinierten Schlüsselspeichers. Sie finden die ID auf der Seite Benutzerdefinierte Schlüsselspeicher in der -Konsole oder mithilfe der -DescribeCustomKeyStoresOperation ohne Parameter. Ersetzen Sie vor Ausführung dieses Beispiels die Beispiel-ID durch eine gültige ID.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Um zu überprüfen, ob der -AWS CloudHSMSchlüsselspeicher verbunden ist, verwenden Sie die -DescribeCustomKeyStoresOperation. Diese Produktion gibt standardmäßig alle benutzerdefinierten Schlüsselspeicher innerhalb des Kontos und der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die

Antwort auf bestimmte benutzerdefinierte Schlüsselspeicher zu begrenzen. Der `ConnectionState`-Wert `CONNECTED` gibt an, dass der benutzerdefinierte Schlüsselspeicher mit seinem AWS CloudHSM-Cluster verbunden ist.

> **Note**
>
> Das Feld `CustomKeyStoreType` wurde der Antwort `DescribeCustomKeyStores` hinzugefügt, um AWS CloudHSM-Schlüsselspeicher von externen Schlüsselspeichern zu unterscheiden.

```
$ 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"
   ],
}
```

Wenn der `ConnectionState`-Wert „Failed (Fehlgeschlagen)" lautet, ist im Element `ConnectionErrorCode` die Fehlerursache angegeben. In diesem Fall konnte AWS KMS in Ihrem Konto keinen AWS CloudHSM-Cluster mit der Cluster-ID `cluster-1a23b4cdefg` finden. Wenn Sie den Cluster gelöscht haben, können Sie ihn von einer Sicherung des ursprünglichen Clusters wiederherstellen und dann die Cluster-ID für den benutzerdefinierten Schlüsselspeicher bearbeiten. Hilfe beim Beantworten eines Verbindungsfehlercodes finden Sie unter Beheben eines Verbindungsfehlers.

```
$ 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"
   ],
}
```

Danach: Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher.

Trennen der Verbindung für einen AWS CloudHSM-Schlüsselspeicher (Konsole)

Um die Verbindung eines verbundenen AWS CloudHSM-Schlüsselspeichers in der AWS Management Console zu trennen, wählen Sie den AWS CloudHSM-Schlüsselspeicher zunächst auf der Seite Custom Key Stores (Benutzerdefinierte Schlüsselspeicher) aus.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.
4. Wählen Sie die Zeile des externen Schlüsselspeichers aus, mit dem Sie die Verbindung trennen möchten.
5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Disconnect (Verbindung trennen) aus.

Nach Abschluss der Produktion ändert sich der Verbindungsstatus von Verbindung wird getrennt in Verbindung wird getrennt. Wenn die Produktion fehlschlägt, wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung für einen Custom Key Store.

Trennen der Verbindung für einen AWS CloudHSM-Schlüsselspeicher (API)

Um die Verbindung eines verbundenen AWS CloudHSM Schlüsselspeichers zu trennen, verwenden Sie die -DisconnectCustomKeyStoreOperation. Wenn die Produktion erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

In diesem Beispiel wird die Verbindung eines AWS CloudHSM-Schlüsselspeichers getrennt. Ersetzen Sie vor Ausführung dieses Beispiels die Beispiel-ID durch eine gültige ID.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Um zu überprüfen, ob der -AWS CloudHSMSchlüsselspeicher getrennt ist, verwenden Sie die -DescribeCustomKeyStoresOperation. Diese Produktion gibt standardmäßig alle benutzerdefinierten Schlüsselspeicher innerhalb des Kontos und der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die Antwort auf bestimmte benutzerdefinierte Schlüsselspeicher zu begrenzen. Der `ConnectionState`-Wert `DISCONNECTED` gibt an, dass der AWS CloudHSM-Schlüsselspeicher nicht mit seinem AWS CloudHSM-Cluster verbunden ist.

```
$ 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>"
   ],
}
```

Löschen eines AWS CloudHSM-Schlüsselspeichers

Wenn Sie einen AWS CloudHSM-Schlüsselspeicher löschen, löscht AWS KMS alle Metadaten zum AWS CloudHSM-Schlüsselspeicher aus KMS, einschließlich Informationen zu dessen Verknüpfung mit einem AWS CloudHSM-Cluster. Diese Produktion hat keine Auswirkungen auf den AWS CloudHSM-Cluster, dessen HSMs oder dessen Benutzer. Sie können einen neuen AWS CloudHSM-Schlüsselspeicher erstellen, der mit demselben AWS CloudHSM-Cluster verknüpft ist. Sie können den Löschvorgang jedoch nicht rückgängig machen.

Sie können ausschließlich AWS CloudHSM-Schlüsselspeicher löschen, die von ihrem AWS CloudHSM-Cluster getrennt wurden und keine AWS KMS keys enthalten. Vor dem Löschen eines benutzerdefinierten Schlüsselspeichers müssen Sie folgende Schritte ausführen.

- Stellen Sie sicher, dass Sie keinen der KMS-Schlüssel in dem Schlüsselspeicher für kryptografische Produktionen benötigen. Anschließend planen Sie die Löschung aller KMS-Schlüssel aus dem Schlüsselspeicher. Informationen dazu, wie Sie die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher ermitteln, finden Sie unter Ermitteln der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher.
- Überprüfen Sie, ob alle KMS-Schlüssel gelöscht wurden. Informationen zum Anzeigen der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher finden Sie unter Anzeigen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher.
- Trennen Sie den AWS CloudHSM-Schlüsselspeicher von dessen AWS CloudHSM-Cluster.

Sie sollten anstelle des Löschens des AWS CloudHSM-Schlüsselspeichers dessen Trennung vom verknüpften AWS CloudHSM-Cluster in Betracht ziehen. Sie können den AWS CloudHSM-Schlüsselspeicher und dessen AWS KMS keys verwalten, während der AWS CloudHSM-Schlüsselspeicher getrennt ist. Sie können jedoch keine KMS-Schlüssel im AWS CloudHSM-Schlüsselspeicher erstellen oder verwenden. Sie können die Verbindung des AWS CloudHSM-Schlüsselspeichers jederzeit wiederherstellen.

Themen

- Löschen eines AWS CloudHSM Schlüsselsspeichers (Konsole)
- Löschen eines AWS CloudHSM-Schlüsselspeichers (API)

Löschen eines AWS CloudHSM Schlüsselsspeichers (Konsole)

Um einen AWS CloudHSM-Schlüsselspeicher in der AWS Management Console zu löschen, wählen Sie zunächst den AWS CloudHSM-Schlüsselspeicher auf der Seite Custom key stores (Benutzerdefinierte Schlüsselspeicher) aus.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (benutzerdefinierte Schlüsselspeicher) AWS CloudHSM-Schlüsselspeicher aus.
4. Suchen Sie die Zeile, die den AWS CloudHSM-Schlüsselspeicher darstellt, den Sie löschen möchten. Wenn der Verbindungsstatus des AWS CloudHSM-Schlüsselspeichers nicht

Verbindung getrennt ist, müssen Sie die Verbindung des AWS CloudHSM-Schlüsselspeichers trennen, bevor Sie ihn löschen.

5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Delete (Löschen) aus.

Nach Abschluss des Vorgangs wird eine Erfolgsmeldung angezeigt und der AWS CloudHSM-Schlüsselspeicher wird nicht mehr in der Schlüsselspeicherliste angezeigt. Wenn die Produktion nicht erfolgreich ist, wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung für einen Custom Key Store.

Löschen eines AWS CloudHSM-Schlüsselspeichers (API)

Um einen -AWS CloudHSMSchlüsselspeicher zu löschen, verwenden Sie die -DeleteCustomKeyStoreOperation. Wenn die Produktion erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück.

Überprüfen Sie zunächst, dass der AWS CloudHSM-Schlüsselspeicher keine AWS KMS keys enthält. Sie können benutzerdefinierte Schlüsselspeicher, die KMS-Schlüssel enthalten, nicht löschen. Der erste Beispielbefehl verwendet ListKeys und DescribeKey, um AWS KMS keys im -AWS CloudHSMSchlüsselspeicher nach mit der Beispiel-ID des benutzerdefinierten Schlüsselspeichers *cks-1234567890abcdef0* zu suchen. In diesem Fall gibt der Befehl keine KMS-Schlüssel zurück. Wenn dies der Fall ist, verwenden Sie die -ScheduleKeyDeletionOperation, um das Löschen der einzelnen KMS-Schlüssel zu planen.

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'
```

Als Nächstes trennen Sie den AWS CloudHSM-Schlüsselspeicher. Dieser Beispielbefehl verwendet die -DisconnectCustomKeyStoreOperation, um einen -AWS CloudHSMSchlüsselspeicher von seinem

-AWS CloudHSMCluster zu trennen. Vor der Ausführung dieses Befehls müssen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers durch eine gültige ID ersetzen.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Nachdem der benutzerdefinierte Schlüsselspeicher getrennt wurde, können Sie ihn mit der DeleteCustomKeyStore Operation löschen.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Verwalten von KMS-Schlüsseln in einem CloudHSM-Schlüsselspeicher

Sie können die AWS KMS keys in einem AWS CloudHSM-Schlüsselspeicher erstellen, anzeigen, verwalten, verwenden und deren Löschung planen. Die dafür verwendeten Verfahren ähneln denjenigen für andere KMS-Schlüssel. Der einzige Unterschied besteht darin, dass Sie einen AWS CloudHSM-Schlüsselspeicher angeben, wenn Sie den KMS-Schlüssel erstellen. Anschließend erstellt AWS KMS nicht extrahierbares Schlüsselmaterial für den KMS-Schlüssel im AWS CloudHSM-Cluster, der dem AWS CloudHSM-Schlüsselspeicher zugeordnet ist. Wenn Sie einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher verwenden, werden die kryptografischen Vorgänge in den HSMs im Cluster ausgeführt.

Unterstützte Funktionen

Zusätzlich zu den in diesem Abschnitt beschriebenen Verfahren können Sie mit KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher folgende Aktionen ausführen:

- Verwenden Sie Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen zur Autorisierung des Zugriffs auf die KMS-Schlüssel.
- Aktivieren und Deaktivieren der KMS-Schlüssel.
- Zuweisen von Tags und Erstellen von Aliassen und Autorisieren des Zugriffs auf die KMS-Schlüssel mithilfe der attributbasierten Zugriffskontrolle (ABAC)
- Verwenden Sie die KMS-Schlüssel für kryptografische Operationen, einschließlich der Verschlüsselung, Entschlüsselung, erneuten Verschlüsselung und Generierung von Datenschlüsseln.
- Verwenden Sie die KMS-Schlüssel mit AWS-Services, die in AWS KMS integriert werden können und kundenverwaltete KMS-Schlüssel unterstützen.
- Verfolgen Sie die Verwendung Ihrer KMS-Schlüssel in -AWS CloudTrailProtokollen und Amazon-CloudWatch Überwachungstools.

Nicht unterstützte Funktionen

- AWS CloudHSM-Schlüsselspeicher unterstützen nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können in einem AWS CloudHSM-Schlüsselspeicher keine HMAC-KMS-Schlüssel, asymmetrische KMS-Schlüssel oder asymmetrische Datenschlüsselpaare erstellen.
- Sie können in einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher kein Schlüsselmaterial importieren. AWS KMS generiert das Schlüsselmaterial für den KMS-Schlüssel im AWS CloudHSM-Cluster.
- Sie können die automatische Rotation des Schlüsselmaterials für einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher nicht aktivieren oder deaktivieren.

Themen

- Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher
- Anzeigen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher
- Verwenden von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher
- Ermitteln von KMS-Schlüssel und Schlüsselmaterial
- Planen der Löschung von KMS-Schlüsseln aus einem AWS CloudHSM-Schlüsselspeicher

Erstellen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher

Wenn Sie einen AWS CloudHSM-Schlüsselspeicher erstellt haben, können Sie in diesem AWS KMS keys erstellen. Es muss sich dabei um KMS-Schlüssel mit symmetrischer Verschlüsselung mit Schlüsselmaterial, das AWS KMS generiert, handeln. Sie können asymmetrische KMS-Schlüssel, HMAC-KMS-Schlüssel oder KMS-Schlüssel mit importiertem Schlüsselmaterial nicht in einem benutzerdefinierten Schlüsselspeicher erstellen. Außerdem können Sie auch keine KMS-Schlüssel mit symmetrischer Verschlüsselung in einem benutzerdefinierten Schlüsselspeicher verwenden, um asymmetrische Datenschlüsselpaare zu generieren.

Um einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher zu erstellen, muss der AWS CloudHSM-Schlüsselspeicher mit seinem AWS CloudHSM-Cluster verbunden sein, und der Cluster muss über midestens zwei aktive HSMs in verschiedenen Availability Zones verfügen. Sie finden den Verbindungsstatus und die Anzahl der HSMs auf der Seite der AWS CloudHSM-Schlüsselspeicher in der AWS Management Console. Wenn Sie die -API-Operationen verwenden, verwenden Sie die -DescribeCustomKeyStoresOperation, um zu überprüfen, ob der -AWS CloudHSMSchlüsselspeicher verbunden ist. Verwenden Sie die AWS CloudHSM DescribeClusters Operation , um die Anzahl der aktiven HSMs im Cluster und deren Availability Zones zu überprüfen.

Wenn Sie einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher erstellen, erstellt AWS KMS den KMS-Schlüssel in AWS KMS. Das Schlüsselmaterial für den KMS-Schlüssel wird hingegen in dem zugehörigen AWS CloudHSM-Cluster erstellt. Insbesondere meldet sich AWS KMS als der `kmsuser`-Kryptobenutzer an, den Sie erstellt haben, an dem Cluster an. Anschließend erstellt AWS KMS einen persistenten, nicht extrahierbaren, symmetrischen 256-Bit-AES-Schlüssel (Advanced Encryption Standard) in dem Cluster und legt den Wert für das Schlüsselbezeichnungsattribut, das nur innerhalb des Clusters sichtbar ist, auf den Amazon-Ressourcenname (ARN) des KMS-Schlüssel fest.

Wenn der Befehl erfolgreich ausgeführt wurde, lautet der Schlüsselstatus des neuen KMS-Schlüssels `Enabled` und der Ursprung `AWS_CLOUDHSM`. Der Ursprung eines KMS-Schlüssels kann nicht mehr geändert werden, nachdem Sie ihn erstellt haben. Wenn Sie einen KMS-Schlüssel in einem -AWS CloudHSMSchlüsselspeicher in der -AWS KMSKonsole oder mithilfe der -DescribeKeyOperation anzeigen, können Sie typische Eigenschaften wie Schlüssel-ID, Schlüsselstatus und Erstellungsdatum sehen. Daneben können Sie aber auch die ID des benutzerdefinierten Schlüsselspeichers sowie (optional) die AWS CloudHSM-Cluster-ID. Details hierzu finden Sie unter Anzeigen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher.

Wenn bei dem Versuch, einen KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher zu erstellen, ein Fehler auftritt, nutzen Sie die Fehlermeldung, um die Ursache des Problems zu

ermitteln. Die Fehlermeldungen können beispielsweise darauf hinweisen, dass der AWS CloudHSM-Schlüsselspeicher nicht verbunden ist (`CustomKeyStoreInvalidStateException`), oder dass der zugehörige AWS CloudHSM-Cluster nicht über die erforderlichen zwei HSMs für diese Produktion verfügt (`CloudHsmClusterInvalidConfigurationException`). Weitere Informationen dazu finden Sie unter Fehlerbehebung für einen Custom Key Store.

Ein Beispiel des AWS CloudTrail-Protokolls der Produktion, die einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher erstellt, finden Sie unter CreateKey.

Themen

- Erstellen eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher (Konsole)
- Erstellen eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher (API)

Erstellen eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher (Konsole)

Führen Sie die folgenden Schritte aus, um einen KMS-Schlüssel mit symmetrischer Verschlüsselung in einem AWS CloudHSM-Schlüsselspeicher zu erstellen.

> **Note**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Wählen Sie Symmetric (Symmetrisch).
6. Unter Key usage (Schlüsselverwendung) ist die Option Encrypt and decrypt (Verschlüsseln und Entschlüsseln) für Sie ausgewählt. Ändern Sie dies nicht.
7. Wählen Sie Advanced options (Erweiterte Optionen) aus.

8. Wählen Sie unter Ursprung des Schlüsselmaterials die Option AWS CloudHSM-Schlüsselspeicher aus.

   Sie können keine multiregionale Schlüssel in einem AWS CloudHSM-Schlüsselspeicher verwenden.

9. Wählen Sie Weiter aus.

10. Wählen Sie einen AWS CloudHSM-Schlüsselspeicher für den neuen KMS-Schlüssel aus. Wählen Sie zum Erstellen des neuen AWS CloudHSM-Schlüsselspeichers die Option Create custom key store (Benutzerdefinierten Schlüsselspeicher erstellen).

    Der ausgewählte AWS CloudHSM-Schlüsselspeicher muss den Status Verbunden haben. Der zugehörige AWS CloudHSM-Cluster muss aktiv sein und über mindestens zwei aktive HSMs in verschiedenen Availability Zones verfügen.

    Hilfe zum Herstellen der Verbindung mit einem AWS CloudHSM-Schlüsselspeicher finden Sie unter Herstellen und Trennen der Verbindung eines AWS CloudHSM-Schlüsselspeichers. Hilfe zum Hinzufügen von HSMs finden Sie unter Hinzufügen eines HSM im AWS CloudHSM-Benutzerhandbuch.

11. Wählen Sie Weiter aus.

12. Geben Sie einen Alias und eine optionale Beschreibung für den KMS-Schlüssel ein.

13. (Optional). Fügen Sie auf der Seite Add Tags (Tags hinzufügen) Tags hinzu, um Ihre KMS-Schlüssel identifizieren zu können und sie zu kategorisieren.

    Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

14. Wählen Sie Weiter aus.

15. Sie können im Abschnitt Key administrators (Schlüsseladministratoren) die IAM-Benutzer und -Rollen auswählen, die den KMS-Schlüssel verwalten dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsseladministratoren die Verwaltung des KMS-Schlüssels.

> **Note**
>
> Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssel auch über IAM-Richtlinien erhalten.

> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

16. (Optional) Um zu verhindern, dass diese Schlüsseladministratoren diesen KMS-Schlüssel löschen, deaktivieren Sie das Kontrollkästchen unten auf der Seite neben Allow key administrators to delete this key (Administratoren erlauben, diesen Schlüssel zu löschen).
17. Wählen Sie Weiter aus.
18. Wählen Sie im Abschnitt This account (dieses Konto) die IAM-Benutzer und -Rollen in diesem AWS-Konto aus, die den KMS-Schlüssel für kryptografische Operationen verwenden dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssels.

> **Note**
>
> Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssels auch über IAM-Richtlinien erhalten.
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

19. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other (Andere Konten) AWS-Konten unten auf der Seite die Option Add another (Weiteres Konto hinzufügen)AWS-Konto aus und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

> **Note**
>
> Administratoren der anderen AWS-Konten müssen auch Zugriff auf den KMS-Schlüssel erlauben, indem Sie IAM-Richtlinien für ihre Benutzer erstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

20. Wählen Sie Weiter.

21. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
22. Wählen Sie danach Finish (Fertigstellen) aus.

Wenn der Vorgang erfolgreich abgeschlossen wird, wird der neue KMS-Schlüssel in dem ausgewählten AWS CloudHSM-Schlüsselspeicher angezeigt. Wenn Sie den Namen oder Alias des neuen KMS-Schlüssels auswählen, werden auf der Registerkarte Cryptographic configuration (Kryptografische Konfiguration) auf der Detailseite der Ursprung des KMS-Schlüssels (AWS CloudHSM), der Name, die ID und der Typ des benutzerdefinierten Schlüsselspeichers sowie die ID, des AWS CloudHSM-Clusters angezeigt. Wenn der Vorgang fehlschlägt, wird unter eine Fehlermeldung mit einer Beschreibung des Fehlers angezeigt.

> **Tip**
>
> Sie können die KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher leichter identifizieren, wenn Sie auf der Seite Customer managed keys (kundenverwaltete Schlüssel) festlegen, dass die Spalte Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) angezeigt wird. Klicken Sie auf das Zahnradsymbol rechts oben und wählen Sie dann Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) aus. Details hierzu finden Sie unter Anpassen Ihrer KMS-Schlüsseltabellen.

Erstellen eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher (API)

Um einen neuen AWS KMS key (KMS-Schlüssel) in Ihrem AWS CloudHSM-Schlüsselspeicher zu erstellen, verwenden Sie die -CreateKeyOperation. Verwenden Sie den Parameter `CustomKeyStoreId`, um den benutzerdefinierten Schlüsselspeicher zu identifizieren und den `Origin`-Wert auf `AWS_CLOUDHSM` festzulegen.

Gegebenenfalls können Sie auch mit dem Parameter `Policy` eine Schlüsselrichtlinie angeben. Sie können die Schlüsselrichtlinie (PutKeyPolicy) ändern und optionale Elemente wie eine Beschreibung und Tags jederzeit hinzufügen.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Das folgende Beispiel beginnt mit einem Aufruf der -DescribeCustomKeyStoresOperation, um zu überprüfen, ob der -AWS CloudHSMSchlüsselspeicher mit dem zugehörigen -AWS

CloudHSMCluster verbunden ist. Diese Produktion gibt standardmäßig alle benutzerdefinierten Schlüsselspeicher innerhalb des Kontos und der Region zurück. Wenn Sie die Daten für nur einen bestimmten AWS CloudHSM-Schlüsselspeicher zurückgeben möchten, verwenden Sie seine Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (die aber nicht miteinander kombiniert werden können).

Wenn Sie diesen Befehl ausführen, denken Sie daran, die ID des benutzerdefinierten Schlüsselspeichers in dem Beispiel durch eine gültige ID zu ersetzen.

> **Note**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

```
$ 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"
   ],
}
```

Der nächste Beispielbefehl verwendet die -DescribeClustersOperation, um zu überprüfen, ob der AWS CloudHSM Cluster, der dem zugeordnet ist `ExampleKeyStore` (cluster-1a23b4cdefg), über mindestens zwei aktive HSMs verfügt. Wenn der Cluster weniger als zwei HSMs hat, schlägt die `CreateKey`-Produktion fehl.

```
$ 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"
        }
    ]
}
```

Dieser Beispielbefehl verwendet die -CreateKeyOperation, um einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher zu erstellen. Um einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher zu erstellen, müssen Sie die ID des AWS CloudHSM-Schlüsselspeichers angeben und als `Origin`-Wert `AWS_CLOUDHSM` festlegen.

Die Antwort enthält u.a. die ID des benutzerdefiniertes Schlüsselspeichers und den AWS CloudHSM-Cluster.

Wenn Sie diesen Befehl ausführen, denken Sie daran, die ID des benutzerdefinierten Schlüsselspeichers in dem Beispiel durch eine gültige ID zu ersetzen.

```
$ 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"
    ]
  }
}
```

Anzeigen von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher

Um einen AWS KMS keys in einem AWS CloudHSM-Schlüsselspeicher anzuzeigen, verwenden Sie denselben Vorgang, wie für alle anderen kundenverwalteten AWS KMS-Schlüssel. Informationen zu den Grundlagen finden Sie unter Anzeigen von Schlüsseln. Informationen zur Identifizierung der Schlüssel in Ihrem AWS CloudHSM-Cluster, die als Schlüsselmaterial für Ihren KMS-Schlüssel dienen, finden Sie unter Ermitteln von KMS-Schlüssel und Schlüsselmaterial. Weitere Informationen zur Anzeige von AWS CloudTrail-Protokollen, die alle API-Operationen einem benutzerdefiniertem Schlüsselspeicher aufzeichnen, finden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

Die KMS-Schlüssel in Ihrem benutzerdefinierten Schlüsselspeicher werden in der AWS KMS-Konsole zusammen mit allen anderen kundenverwalteten Schlüsseln in Ihrem AWS-Konto und Ihrer Region angezeigt.

Die folgenden Werte sind jedoch für KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher spezifisch.

- Der Name und die ID des AWS CloudHSM-Schlüsselspeichers, in dem der KMS-Schlüssel gespeichert ist.
- Die Cluster-ID des zugehörigen AWS CloudHSM-Clusters, das das Schlüsselmaterial enthält.
- Der Wert `Origin` `AWS CloudHSM` in der AWS KMS-Konsole oder `AWS_CLOUDHSM` in API-Antworten.
- Der Wert für den Schlüsselstatus kann `Unavailable` sein. Informationen dazu, wie Sie den Status auflösen, finden Sie unter So reparieren Sie nicht-verfügbare KMS-Schlüssel.

Anzeigen der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher (Konsole)

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie oben rechts das Zahnradsymbol und dann Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) und Origin (Ursprung) aus. Wählen Sie anschließend Confirm (Bestätigen) aus.
5. Um KMS-Schlüssel in allen AWS CloudHSM-Schlüsselspeichern zu identifizieren, suchen Sie nach KMS-Schlüsseln mit dem Wert AWS CloudHSM für Origin (Ursprung). Um KMS-Schlüssel in einem bestimmten AWS CloudHSM-Schlüsselspeicher zu identifizieren, zeigen Sie die Werte in der Spalte Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) an.
6. Wählen Sie den Alias oder die Schlüssel-ID eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher aus.

   Auf dieser Seite werden detaillierte Informationen zum KMS-Schlüssel angezeigt, einschließlich dessen Amazon-Ressourcenname (ARN), Schlüsselrichtlinie und Tags.
7. Wählen Sie die Registerkarte Cryptographic configuration (kryptografische Konfiguration) aus. Die Registerkarte wird unter dem Abschnitt General Configuration (allgemeine Konfiguration) angezeigt.

Dieser Bereich enthält Informationen zum AWS CloudHSM-Schlüsselspeicher und AWS CloudHSM-Cluster des KMS-Schlüssels.

Anzeigen der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher (API)

Sie verwenden dieselben AWS KMS API-Operationen, um die KMS-Schlüssel in einem ListKeys-AWS CloudHSMSchlüsselspeicher anzuzeigen, die Sie für jeden KMS-Schlüssel verwenden würden, einschließlich DescribeKey, und GetKeyPolicy. Beispielsweise zeigt der folgende `describe-key`-Vorgang in der AWS CLI die speziellen Felder für einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher an. Vor der Ausführung eines Befehls wie diesem müssen Sie die ID des Beispiel-KMS-Schlüssels durch einen gültigen Wert ersetzen.

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

{
   "KeyMetadata": {
      "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "AWSAccountId": "111122223333",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CreationDate": 1537582718.431,
      "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "Description": "Key in custom key store",
      "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"
   }
}
```

Weitere Informationen dazu, wie Sie die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher finden oder die Schlüssel in Ihrem AWS CloudHSM-Cluster identifizieren, die als

Schlüsselmaterial für Ihren KMS-Schlüssel dienen, finden Sie unter Ermitteln von KMS-Schlüssel und Schlüsselmaterial.

Verwenden von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher

Nachdem Sie einen KMS-Schlüssel mit symmetrischer Verschlüsselung in einem AWS CloudHSMSchlüsselspeicher erstellt haben, können Sie ihn für die folgenden kryptografischen Vorgänge verwenden:

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Die Operationen, die asymmetrische Datenschlüsselpaare generieren, GenerateDataKeyPair und GenerateDataKeyPairWithoutPlaintext, werden in benutzerdefinierten Schlüsselspeichern nicht unterstützt.

Wenn Sie den KMS-Schlüssel in einer Anforderung verwenden, identifizieren Sie den KMS-Schlüssel anhand seiner ID oder seines Alias. Sie müssen nicht den AWS CloudHSM-Schlüsselspeicher oder AWS CloudHSM-Cluster angeben. Die Antwort enthält die gleichen Felder, die auch für alle anderen KMS-Schlüssel mit symmetrischer Verschlüsselung zurückgegeben werden.

Wenn Sie jedoch einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher verwenden, wird der kryptografische Vorgang vollständig innerhalb des AWS CloudHSM-Clusters ausgeführt, der dem AWS CloudHSM-Schlüsselspeicher zugeordnet ist. Die Produktion verwendet das Schlüsselmaterial im Cluster, das dem ausgewählten KMS-Schlüssel zugeordnet ist.

Damit dies möglich ist, sind die folgenden Bedingungen erforderlich.

- Der Schlüsselstatus des KMS-Schlüssels muss `Enabled` lauten. Um den Schlüsselstatus zu finden, verwenden Sie das Feld Status in der AWS KMSKonsole oder das `KeyState` Feld in der DescribeKey Antwort.
- Der AWS CloudHSM-Schlüsselspeicher muss mit seinem AWS CloudHSM-Cluster verbunden sein. Der Status in der AWS KMS Konsole oder `ConnectionState` in der DescribeCustomKeyStores Antwort muss lauten`CONNECTED`.

- Der dem benutzerdefinierten Schlüsselspeicher zugeordnete AWS CloudHSM-Cluster muss mindestens ein aktives HSM enthalten. Um die Anzahl der aktiven HSMs im Cluster zu ermitteln, verwenden Sie die AWS KMS -Konsole, die -AWS CloudHSMKonsole oder die -DescribeClustersOperation.
- Der AWS CloudHSM-Cluster muss das Schlüsselmaterial für den KMS-Schlüssel enthalten. Wenn das Schlüsselmaterial aus dem Cluster gelöscht wurden oder ein HSM aus einer Sicherung erstellt wurde, in der die Schlüsselinformationen nicht enthalten waren, schlägt die kryptografische Produktion fehl.

Wenn diese Bedingungen nicht erfüllt sind, schlägt die kryptografische Produktion fehl und AWS KMS gibt die Ausnahme `KMSInvalidStateException` zurück. In der Regel müssen Sie einfach nur den AWS CloudHSM-Schlüsselspeicher erneut verbinden. Weitere Informationen finden Sie unter Beheben eines fehlerhaften KMS-Schlüssels.

Bei der Verwendung von KMS-Schlüsseln in einem AWS CloudHSM-Schlüsselspeicher ist zu beachten, dass die KMS-Schlüssel in jedem AWS CloudHSM-Schlüsselspeicher gemeinsam ein Anforderungskontingent für benutzerdefinierte Schlüsselspeicher für kryptografische Vorgänge nutzen. Wenn Sie das Kontingent überschreiten, gibt AWS KMS `ThrottlingException` zurück. Wenn der dem AWS CloudHSM-Schlüsselspeicher zugeordnete AWS CloudHSM-Cluster zahlreiche Befehle verarbeitet, einschließlich solcher, die unabhängig vom AWS CloudHSM-Schlüsselspeicher sind, kann bereits bei einer geringeren Rate eine `ThrottlingException` auftreten. Wenn Sie eine `ThrottlingException` für eine Anforderung erhalten, verringern Sie Ihre Anforderungsrate und führen Sie die Befehle erneut aus. Details zum Anforderungskontingent für benutzerdefinierte Schlüsselspeicher finden Sie unter Anforderungskontingente für benutzerdefinierte Schlüsselspeicher.

Ermitteln von KMS-Schlüssel und Schlüsselmaterial

Bei der Verwaltung eines AWS CloudHSM-Schlüsselspeichers müssen Sie möglicherweise die KMS-Schlüssel in den einzelnen AWS CloudHSM-Schlüsselspeichern ermitteln. Beispielsweise könnte es sein, dass Sie eine der folgenden Aufgaben ausführen müssen:

- Nachverfolgung der KMS-Schlüssel im AWS CloudHSM-Schlüsselspeicher in AWS CloudTrail-Protokollen
- Die Auswirkung der Trennung eines AWS CloudHSM-Schlüsselspeichers auf KMS-Schlüssel prognostizieren.

- Einplanen des Löschens von KMS-Schlüsseln vor dem Löschen eines AWS CloudHSM-Schlüsselspeichers

Vielleicht möchten Sie auch die Schlüssel in Ihrem AWS CloudHSM-Cluster ermitteln, die als Schlüsselmaterial für Ihre KMS-Schlüssel dienen. Zwar verwaltet AWS KMS die KMS-Schlüssel und ihr Schlüsselmaterial, Sie behalten jedoch die Kontrolle über Ihren AWS CloudHSM-Cluster, seine HSMs und Sicherungen sowie die Schlüssel in den HSMs und sind weiterhin für deren Verwaltung zuständig. Möglicherweise müssen Sie die Schlüssel ermitteln, um das Schlüsselmaterial zu prüfen, es vor versehentlichem Löschen zu schützen oder es nach dem Löschen des KMS-Schlüssels aus HSMs und Cluster-Sicherungen zu löschen.

Alle Schlüsselinformationen für die KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher sind Eigentum des `kmsuser`-Kryptobenutzers (CU). AWS KMS legt für das Attribut für die Schlüsselbezeichnung, das nur in AWS CloudHSM angezeigt werden kann, den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels fest.

Wenden Sie zum Suchen der KMS-Schlüssel und Schlüsselmaterial eine der folgenden Methoden an.

- Ermitteln der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher – Ermitteln der KMS-Schlüssel in einem oder allen Ihren AWS CloudHSM-Schlüsselspeichern.
- Finden aller Schlüssel für einen AWS CloudHSM-Schlüsselspeicher – Ermitteln aller Schlüssel in Ihrem Cluster, die als Schlüsselmaterial für die KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher dienen.
- Ermitteln des AWS CloudHSM-Schlüssels für einen KMS-Schlüssel – Ermitteln des Schlüssels in Ihrem Cluster, der als Schlüsselmaterial für einen bestimmten KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher dient.
- Ermitteln des KMS-Schlüssels für einen AWS CloudHSM-Schlüssel – Ermitteln des KMS-Schlüssels für einen bestimmten Schlüssel in Ihrem Cluster.

Ermitteln der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher

Bei der Verwaltung eines AWS CloudHSM-Schlüsselspeichers müssen Sie möglicherweise die KMS-Schlüssel in den einzelnen AWS CloudHSM-Schlüsselspeichern ermitteln. Mithilfe dieser Informationen können Sie die KMS-Schlüssel-Produktionen in AWS CloudTrail-Protokollen nachverfolgen, die Auswirkung der Trennung eines benutzerdefinierten Schlüsselspeichers auf die

KMS-Schlüssel prognostizieren oder das Löschen von KMS-Schlüssel vor dem Löschen eines AWS CloudHSM-Schlüsselspeichers einplanen.

Ermitteln der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher (Konsole)

Zeigen Sie zum Ermitteln der KMS-Schlüssel in einem bestimmten AWS CloudHSM-Schlüsselspeicher auf der Seite Customer managed keys (kundenverwaltete Schlüssel) die Werte in den Feldern Custom Key Store Name (Name des benutzerdefinierten Schlüsselspeichers) oder Custom Key Store ID (ID des benutzerdefinierten Schlüsselspeichers) an. Um KMS-Schlüssel in allen AWS CloudHSM-Schlüsselspeichern zu identifizieren, suchen Sie nach KMS-Schlüsseln mit dem Wert AWS CloudHSM für Origin (Ursprung). Wenn Sie der Anzeige optionale Spalten hinzufügen möchten, wählen Sie das Zahnradsymbol rechts oben auf der Seite aus.

Ermitteln der KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher (API)

Um die KMS-Schlüssel in einem -AWS CloudHSMSchlüsselspeicher zu finden, verwenden Sie die DescribeKey Operationen ListKeys und und filtern Sie dann nach `CustomKeyStoreId` Wert. Bevor Sie die Beispiele ausführen, ersetzen Sie die fiktiven Werte für die ID des benutzerdefinierten Schlüsselspeichers durch einen gültigen Wert.

Bash

Wenn Sie die KMS-Schlüssel in einem bestimmten AWS CloudHSM-Schlüsselspeicher suchen, rufen Sie alle Ihre KMS-Schlüssel in dem Konto und der Region ab. Filtern Sie anschließend nach der ID des benutzerdefinierten Schlüsselspeichers.

```
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
```

Wenn Sie die KMS-Schlüssel in beliebigen AWS CloudHSM-Schlüsselspeichern in dem Konto und der Region abrufen möchten, suchen Sie nach `CustomKeyStoreType`-Werten, die mit `AWS_CloudHSM` beginnen.

```
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

Um KMS-KmsKeyListSchlüssel in einem bestimmten AWS CloudHSM Schlüsselspeicher zu finden, verwenden Sie die Cmdlets Get- und Get-KmsKey, um alle Ihre KMS-Schlüssel im Konto und in der Region abzurufen. Filtern Sie anschließend nach der ID des benutzerdefinierten Schlüsselspeichers.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

Um KMS-Schlüssel in einem beliebigen AWS CloudHSM Schlüsselspeicher im Konto und in der Region abzurufen, filtern Sie nach dem CustomKeyStoreType Wert von `AWS_CLOUDHSM`.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

Finden aller Schlüssel für einen AWS CloudHSM-Schlüsselspeicher

Sie können die Schlüssel in Ihrem AWS CloudHSM-Cluster ermitteln, die als Schlüsselmaterial für Ihren AWS CloudHSM-Schlüsselspeicher dienen. Verwenden Sie dazu den findAllKeys Befehl in cloudhsm_mgmt_util, um die Schlüsselhandles aller Schlüssel zu finden, die `kmsuser` besitzt oder gemeinsam nutzen. Wenn Sie sich nicht als `kmsuser` angemeldet haben und Schlüssel außerhalb von AWS KMS erstellt haben, stellen alle Schlüssel im Besitz von `kmsuser` Schlüsselmaterial für KMS-Schlüssel dar.

Jeder Verschlüsselungsverantwortliche im Cluster kann diesen Befehl ausführen, ohne den AWS CloudHSM-Schlüsselspeicher zu trennen.

1. Starten Sie cloudhsm_mgmt_util wie im Abschnitt Erste Schritte mit CloudHSM Management Utility (CMU) des -Benutzerhandbuchs beschrieben.
2. Melden Sie sich bei cloudhsm_mgmt_util unter Verwendung des Kontos eines Verschlüsselungsverantwortlichen (Crypto Officer, CO) an.
3. Verwenden Sie den Befehl listUsers, um die Benutzer-ID des `kmsuser`-Kryptobenutzers zu finden.

   In diesem Beispiel hat `kmsuser` die Benutzer-ID 3.

   ```
   aws-cloudhsm> listUsers
   ```

```
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt          2FA
         1              PCO             admin                    NO
 0               NO
         2              AU              app_user                 NO
 0               NO
         3              CU              kmsuser                  NO
 0               NO
```

4. Verwenden Sie den findAllKeys Befehl , um die Schlüssel-Handles aller Schlüssel zu finden, die `kmsuser` besitzt oder gemeinsam nutzen. Ersetzen Sie die Beispielbenutzer-ID (3) durch die tatsächliche Benutzer-ID von `kmsuser` in Ihrem Cluster.

   Die Beispielausgabe zeigt, dass `kmsuser` Besitzer von Schlüsseln mit den Schlüssel-Handles 8, 9 und 262162 in beiden HSMs im Cluster ist.

```
aws-cloudhsm> findAllKeys 3 0
Keys on server 0(10.0.0.1):
Number of keys found 3
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 0(10.0.0.1)

Keys on server 1(10.0.0.2):
Number of keys found 6
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 1(10.0.0.2)
```

Ermitteln des KMS-Schlüssels für einen AWS CloudHSM-Schlüssel

Wenn Sie das Schlüssel-Handle eines Schlüssels in dem Cluster kennen, dessen Eigentümer `kmsuser` ist, können Sie anhand der Schlüsselmarkierung den zugehörigen KMS-Schlüssel in Ihrem AWS CloudHSM-Schlüsselspeicher ermitteln.

Wenn AWS KMS das Schlüsselmaterial für einen KMS-Schlüssel in Ihrem AWS CloudHSM-Cluster erstellt, wird der Amazon-Ressourcenname (ARN) des KMS-Schlüssels in die Schlüsselbezeichnung geschrieben. Wenn Sie den Bezeichnungswert nicht geändert haben, können Sie mit dem

getAttribute-Befehl in key_mgmt_util oder cloudhsm_mgmt_util den Schlüssel seinem KMS-Schlüssel zuweisen.

Zur Ausführung dieses Verfahrens müssen Sie den AWS CloudHSM-Schlüsselspeicher vorübergehend trennen, damit Sie sich als `kmsuser`-CU anmelden können.

> **Note**
>
> Sämtliche Versuche, KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen oder vorhandene KMS-Schlüssel in kryptografischen Produktionen zu nutzen, schlagen fehl, während der benutzerdefinierte Schlüsselspeicher getrennt ist. Diese Aktion kann verhindern, dass Benutzer vertrauliche Daten speichern und darauf zugreifen.

1. Trennen Sie den AWS CloudHSM-Schlüsselspeicher (sofern nicht bereits geschehen) und melden Sie sich anschließend als `kmsuser` in key_mgmt_util an, wie unter Trennen und Anmelden erläutert.
2. Verwenden Sie den `getAttribute`-Befehl in key_mgmt_util oder cloudhsm_mgmt_util, um das Markierungsattribut (`OBJ_ATTR_LABEL`, Attribut `3`) für ein bestimmtes Schlüssel-Handle abzurufen.

   Bei diesem Befehl beispielsweise wird `getAttribute` in cloudhsm_mgmt_util zum Abrufen des Bezeichnungsattributs (Attribut `3`) des Schlüssels mit dem Schlüssel-Handle `262162` verwendet. Die Ausgabe zeigt, dass der Schlüssel `262162` als Schlüsselmaterial für den KMS-Schlüssel mit dem ARN `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` dient. Ersetzen Sie vor Ausführung dieses Befehls das Beispiel-Schlüssel-Handle durch ein gültiges Handle.

   Wenn Sie eine Liste der Schlüsselattribute abrufen möchten, verwenden Sie den listAttributes-Befehl oder beachten Sie die Schlüsselattribut-Referenz im AWS CloudHSM-Benutzerhandbuch.

   ```
   aws-cloudhsm> getAttribute 262162 3

   Attribute Value on server 0(10.0.1.10):
   OBJ_ATTR_LABEL
   arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
   ```

3. Melden Sie sich von key_mgmt_util oder cloudhsm_mgmt_util ab und stellen Sie die Verbindung des AWS CloudHSM-Schlüsselspeichers wieder her, wie im Abschnitt Abmelden und erneutes Verbinden erläutert.

Ermitteln des AWS CloudHSM-Schlüssels für einen KMS-Schlüssel

Mithilfe der KMS-Schlüssel-ID eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher können Sie den Schlüssel in Ihrem AWS CloudHSM-Cluster ermitteln, der als Schlüsselmaterial dient. Anschließend können Sie das betreffende Schlüssel-Handle zur Ermittlung des Schlüssels in AWS CloudHSM-Client-Befehlen verwenden.

Wenn AWS KMS das Schlüsselmaterial für einen KMS-Schlüssel in Ihrem AWS CloudHSM-Cluster erstellt, wird der Amazon-Ressourcenname (ARN) des KMS-Schlüssels in die Schlüsselbezeichnung geschrieben. Wenn Sie den Bezeichnungswert nicht geändert haben, können Sie den Befehl findKey in key_mgmt_util verwenden, um das Schlüssel-Handle des Schlüsselmaterials für den KMS-Schlüssel abzurufen. Zur Ausführung dieses Verfahrens müssen Sie den AWS CloudHSM-Schlüsselspeicher vorübergehend trennen, damit Sie sich als `kmsuser`-CU anmelden können.

> **Note**
>
> Sämtliche Versuche, KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen oder vorhandene KMS-Schlüssel in kryptografischen Produktionen zu nutzen, schlagen fehl, während der benutzerdefinierte Schlüsselspeicher getrennt ist. Diese Aktion kann verhindern, dass Benutzer vertrauliche Daten speichern und darauf zugreifen.

1. Trennen Sie den AWS CloudHSM-Schlüsselspeicher (sofern nicht bereits geschehen) und melden Sie sich anschließend als `kmsuser` in key_mgmt_util an, wie unter Trennen und Anmelden erläutert.
2. Verwenden Sie den findKey-Befehl in key_mgmt_util, um nach einem Schlüssel zu suchen, dessen Markierung dem ARN eines KMS-Schlüssels in Ihrem AWS CloudHSM-Schlüsselspeicher entspricht. Ersetzen Sie den Beispiel-KMS-Schlüssel-ARN im Wert des `-l`-Parameters (kleingeschriebenes L für Label) durch einen gültigen KMS-Schlüssel-ARN.

   Dieser Befehl sucht beispielsweise den Schlüssel mit einer Markierung, die dem Beispiel-KMS-Schlüssel-ARN `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` entspricht. Die Beispielausgabe zeigt, dass der Schlüssel mit dem Schlüssel-Handle `262162` den angegebenen

KMS-Schlüssel-ARN in seiner Markierung enthält. Sie können nun dieses Schlüssel-Handle in anderen Befehlen von key_mgmt_util verwenden.

```
Command: findKey -l arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
Total number of keys present 1

 number of keys matched from start index 0::1
262162

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

3. Melden Sie sich von key_mgmt_util ab und stellen Sie die Verbindung des benutzerdefinierten Schlüsselspeichers wieder her, wie im Abschnitt Abmelden und erneutes Verbinden erläutert.

Planen der Löschung von KMS-Schlüsseln aus einem AWS CloudHSM-Schlüsselspeicher

Wenn Sie sich sicher sind, dass Sie einen AWS KMS key nicht mehr für kryptografische Operationen benötigen, können Sie die Löschung des KMS-Schlüssels planen. Gehen Sie dazu genau so vor wie beim Planen der Löschung anderer KMS-Schlüssel aus AWS KMS. Achten Sie dabei jedoch zusätzlich darauf, dass Ihr AWS CloudHSM-Schlüsselspeicher verbunden ist, damit AWS KMS das zugehörige Schlüsselmaterial aus dem angehängten AWS CloudHSM-Cluster löschen kann, nachdem die Wartefrist abgelaufen ist.

Sie können die Planung, Stornierung und Löschung des KMS-Schlüssels in Ihren AWS CloudTrail-Protokollen überwachen.

> **Warning**
>
> Das Löschen eines KMS-Schlüssels ist ein potentiell gefährliches Verfahren, wodurch alle Daten, die mit dem KMS-Schlüssel verschlüsselt wurden, nicht wieder entschlüsselt werden können. Bevor Sie das Löschen des KMS-Schlüssels planen, untersuchen Sie die frühere Nutzung des KMS-Schlüssels und erstellen Sie einen Amazon- CloudWatch Alarm, der Sie benachrichtigt, wenn jemand versucht, den KMS-Schlüssel zu verwenden, während er

gelöscht werden soll. Es wird empfohlen, im Zweifelsfall den KMS-Schlüssel zu deaktivieren anstelle ihn zu löschen.

Wenn Sie die Löschung eines KMS-Schlüssels aus einem AWS CloudHSM-Schlüsselspeicher planen, ändert sich der Status des Schlüssels auf Pending deletion (Löschung ausstehend). Der KMS-Schlüssel verbleibt auch während der Wartezeit im Status Pending deletion (Löschung ausstehend), selbst wenn der KMS-Schlüssel nicht mehr verfügbar ist, weil Sie die Verbindung zu dem benutzerdefinierten Schlüsselspeicher getrennt haben. Dies erlaubt es Ihnen, die Löschung des KMS-Schlüssels vor Ablauf der Wartezeit jederzeit abzubrechen.

Nach Ablauf der Wartezeit löscht AWS KMS den KMS-Schlüssel aus AWS KMS. Anschließend versucht AWS KMS unter Verwendung aller verfügbaren Methoden, das Schlüsselmaterial aus dem zugehörigen AWS CloudHSM-Cluster zu löschen. Wenn AWS KMS Schlüsselmaterial nicht löschen kann, beispielsweise, wenn der Schlüsselspeicher von AWS KMS getrennt wurde, müssen Sie möglicherweise verwaistes Schlüsselmaterial manuell aus dem Cluster löschen.

AWS KMS löscht Schlüsselmaterial nicht aus Cluster-Sicherungen. Selbst wenn Sie KMS-Schlüssel aus AWS KMS löschen und alles Schlüsselmaterial aus Ihrem AWS CloudHSM-Cluster löschen, können anhand von Backups erstellte Cluster das gelöschte Schlüsselmaterial noch enthalten. Um Schlüsselmaterial permanent zu löschen, zeigen Sie das Erstelldatum des KMS-Schlüssels an. Löschen Sie dann Cluster-Sicherungen, die das Schlüsselmaterial noch enthalten könnten.

Wenn Sie die Löschung eines KMS-Schlüssels aus einem AWS CloudHSM-Schlüsselspeicher planen, wird der KMS-Schlüssel sofort unbrauchbar (vorbehaltlich einer letztendlichen Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel geschützten Datenschlüsseln verschlüsselt wurden, sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, von denen viele Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Fehlerbehebung für einen Custom Key Store

AWS CloudHSM-Schlüsselspeicher sind so konzipiert, dass sie zuverlässig und widerstandsfähig sind. Es gibt jedoch einige Fehlerzustände, die Sie möglicherweise beheben müssen, um die Betriebsbereitschaft Ihres AWS CloudHSM-Schlüsselspeichers zu wahren.

Themen

- So reparieren Sie nicht-verfügbare KMS-Schlüssel

- Beheben eines fehlerhaften KMS-Schlüssels
- Beheben eines Verbindungsfehlers
- Wie man auf Fehler bei kryptografischen Produktionen reagiert
- Reparieren ungültiger kmsuser-Anmeldeinformationen
- Löschen von verwaistem Schlüsselmaterial
- Wiederherstellen von gelöschtem Schlüsselmaterial für einen KMS-Schlüssel
- Anmeldung als kmsuser

So reparieren Sie nicht-verfügbare KMS-Schlüssel

Der Schlüsselstatus von AWS KMS keys in einem AWS CloudHSM-Schlüsselspeicher ist normalerweise `Enabled`. Wie bei allen KMS-Schlüsseln ändert sich der Schlüsselstatus, wenn Sie die KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher deaktivieren oder sie zur Löschung einplanen. Im Gegensatz zu anderen KMS-Schlüsseln können die KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher aber auch den Schlüsselstatus `Unavailable` haben.

Der Schlüsselstatus `Unavailable` gibt an, dass sich der KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher befindet, der absichtlich getrennt wurde und eventuelle Versuche, die Verbindung wiederherzustellen, fehlgeschlagen sind. Wenn ein KMS-Schlüssel nicht verfügbar ist, können Sie ihn anzeigen und verwalten, ihn jedoch nicht für kryptographische Produktionen verwenden.

Um den Schlüsselstatus eines KMS-Schlüssels zu ermitteln, zeigen Sie auf der Seite Customer managed keys (kundenverwaltete Schlüssel) das Feld Status des KMS-Schlüssels an. Oder verwenden Sie die -DescribeKeyOperation und zeigen Sie das -`KeyState`Element in der Antwort an. Details hierzu finden Sie unter Anzeigen von Schlüsseln.

Die KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher haben den Schlüsselstatus `Unavailable` oder `PendingDeletion`. KMS-Schlüssel, die zur Löschung aus einem benutzerdefinierten Schlüsselspeicher geplant sind, haben den Schlüsselstatus `Pending Deletion`, auch wenn der benutzerdefinierten Schlüsselspeicher getrennt ist. Auf diese Weise können Sie die geplante Löschung des Schlüssels stornieren, ohne dass Sie erneut eine Verbindung mit dem Custom Key Store herstellen müssen.

Um die Nichtverfügbarkeit eines KMS-Schlüssels zu beheben, verbinden Sie den benutzerdefinierten Schlüsselspeicher wieder. Wenn der benutzerdefinierten Schlüsselspeicher wieder verbunden

ist, wechselt der Schlüsselstatus der KMS-Schlüssels im benutzerdefinierten Schlüsselspeicher automatisch wieder zum vorherigen Zustand, etwa zu `Enabled` oder `Disabled`. KMS-Schlüssel, die zur Löschung ausstehen, bleiben im Status `PendingDeletion`. Während das Problem bestehen bleibt, ändert das Aktivieren und Deaktivieren eines nicht-verfügbaren KMS-Schlüssels dessen Schlüsselstatus nicht. Das Aktivieren und Deaktivieren wirkt sich erst aus, wenn der Schlüssel wieder verfügbar ist.

Für Hilfe bei fehlgeschlagenen Verbindungen vgl. Beheben eines Verbindungsfehlers.

Beheben eines fehlerhaften KMS-Schlüssels

Probleme mit der Erstellung und Verwendung von KMS-Schlüssel in AWS CloudHSM-Schlüsselspeichern können von einem Problem mit Ihrem AWS CloudHSM-Schlüsselspeicher, dem dazugehörigen AWS CloudHSM-Cluster, dem KMS-Schlüssel oder dessen Schlüsselmaterial verursacht werden.

Wenn ein AWS CloudHSM-Schlüsselspeicher von seinem AWS CloudHSM-Cluster getrennt wird, ist der Schlüsselstatus der KMS-Schlüssels in dem benutzerdefinierten Schlüsselspeicher `Unavailable`. Alle Anforderungen zum Erstellen von KMS-Schlüsseln in einem getrennten AWS CloudHSM-Schlüsselspeicher führen zu der Ausnahme `CustomKeyStoreInvalidStateException`. Alle Anforderungen zum Verschlüsseln, erneuten Verschlüsseln oder zum Generieren von Datenschlüsseln führen zu der Ausnahme `KMSInvalidStateException`. Verbinden Sie zur Behebung des Problems den AWS CloudHSM-Schlüsselspeicher erneut.

Ihre Versuche, einen KMS-Schlüssel aus einem AWS CloudHSM-Schlüsselspeicher für kryptographische Produktionen zu verwenden, können jedoch fehlschlagen, selbst wenn sein Schlüsselstatus `Enabled` und der Verbindungsstatus des AWS CloudHSM-Schlüsselspeichers `Connected` ist. Dies kann durch eine der folgenden Ursachen bedingt sein.

- Möglicherweise wurde das Schlüsselmaterial für den KMS-Schlüssel von dem zugehörigen AWS CloudHSM-Cluster gelöscht. Um dies zu ermitteln, suchen Sie zuerst den Schlüssel-Handle des Schlüsselmaterials für einen KMS-Schlüssel und versuchen Sie bei Bedarf, das Schlüsselmaterial wiederherzustellen.
- Alle HSMs wurden aus dem AWS CloudHSM-Cluster gelöscht, der mit dem AWS CloudHSM-Schlüsselspeicher verbunden ist. Damit ein KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher in einer kryptographischen Produktion verwendet werden kann, muss dessen AWS CloudHSM-Cluster mindestens ein aktives HSM enthalten. Um die Anzahl und den

Status von HSMs in einem -AWS CloudHSMCluster zu überprüfen, verwenden Sie die -AWS CloudHSMKonsole oder die -DescribeClustersOperation. Um dem Cluster ein HSM hinzuzufügen, verwenden Sie die -AWS CloudHSMKonsole oder die -CreateHsmOperation.

- Der mit dem AWS CloudHSM-Schlüsselspeicher verbundene AWS CloudHSM-Cluster wurde gelöscht. Um das Problem zu beheben, erstellen Sie einen Cluster aus einer Sicherung, die mit dem ursprünglichen Cluster verbunden ist, oder aus einer Sicherung, die zur Erstellung des ursprünglichen Clusters verwendet wurde. Bearbeiten Sie dann die Cluster-ID in den Einstellungen für den Custom Key Store. Anweisungen finden Sie unter Wiederherstellen von gelöschtem Schlüsselmaterial für einen KMS-Schlüssel.
- Der dem benutzerdefinierten Schlüsselspeicher zugeordnete AWS CloudHSM-Cluster verfügte über keine verfügbaren PKCS#11-Sitzungen. Dies tritt normalerweise in Zeiten mit hohem Burst-Verkehr auf, wenn zusätzliche Sitzungen erforderlich sind, um den Datenverkehr zu bedienen. Um auf eine `KMSInternalException` mit einer Fehlermeldung über PKCS#11-Sitzungen zu antworten, gehen Sie zurück und wiederholen Sie die Anfrage.

Beheben eines Verbindungsfehlers

Wenn Sie versuchen, einen AWS CloudHSM-Schlüsselspeicher mit seinem AWS CloudHSM-Cluster zu verbinden, der Vorgang jedoch fehlschlägt, wechselt der Verbindungsstatus des AWS CloudHSM-Schlüsselspeichers zu `FAILED`. Um den Verbindungsstatus eines -AWS CloudHSMSchlüsselspeichers zu ermitteln, verwenden Sie die -AWS KMSKonsole oder die -DescribeCustomKeyStoresOperation.

Alternativ können einige Verbindungsversuche aufgrund leicht zu erkennender Cluster-Konfigurationsfehler fehlschlagen. In diesem Fall lautet der Verbindungsstatus immer noch `DISCONNECTED`. Diese Fehler geben eine Fehlermeldung oder Ausnahme mit einer Begründung für den fehlgeschlagenen Verbindungsversuch zurück. Überprüfen Sie die Beschreibung der Ausnahme und die -Cluster-Anforderungen, beheben Sie das Problem, aktualisieren Sie den AWS CloudHSM-Schlüsselspeicher, sofern erforderlich, und versuchen Sie erneut, eine Verbindung herzustellen.

Wenn der Verbindungsstatus lautet`FAILED`, führen Sie die -DescribeCustomKeyStoresOperation aus und sehen Sie sich das -`ConnectionErrorCode`Element in der Antwort an.

> **Note**
>
> Wenn der Verbindungsstatus eines AWS CloudHSM-Schlüsselspeichers `FAILED` ist, müssen Sie den AWS CloudHSM-Schlüsselspeicher trennen, bevor Sie versuchen, ihn wieder zu

verbinden. Ein AWS CloudHSM-Schlüsselspeicher mit dem Verbindungsstatus `FAILED` kann nicht verbunden werden.

- `CLUSTER_NOT_FOUND` gibt an, dass AWS KMS keinen AWS CloudHSM-Cluster mit der angegebenen Cluster-ID finden kann. Dies kann auftreten, wenn die falsche Cluster-ID für eine API-Produktion bereitgestellt wurde, oder wenn der Cluster gelöscht und nicht ersetzt wurde. Um diesen Fehler zu beheben, überprüfen Sie die Cluster-ID, z. B. mithilfe der AWS CloudHSM Konsole oder der -DescribeClustersOperation. Wenn der Cluster gelöscht wurde, erstellen Sie einen Cluster aus einer möglichst neuen Sicherung des Originals. Trennen Sie dann den AWS CloudHSM-Schlüsselspeicher, bearbeiten Sie die -Cluster-ID-Einstellung des AWS CloudHSM-Schlüsselspeichers, und verbinden Sie den AWS CloudHSM-Schlüsselspeicher wieder mit dem Cluster.
- `INSUFFICIENT_CLOUDHSM_HSMS` zeigt an, dass der zugehörige AWS CloudHSM-Cluster keine HSMs enthält. Zum herstellen einer Verbindung muss der Cluster mindestens über ein HSM verfügen. Verwenden Sie die -DescribeClustersOperation, um die Anzahl der HSMs im Cluster zu ermitteln. Um diesen Fehler zu beheben, fügen Sie mindestens ein HSM dem Cluster hinzu. Wenn Sie mehrere HSMs hinzufügen, sollten Sie sie in verschiedenen Availability Zones erstellen.
- `INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET` zeigt an, dass AWS KMS den AWS CloudHSM-Schlüsselspeicher nicht mit dessenAWS CloudHSM-Cluster verbinden konnte, weil mindestens ein privates Subnetz, das mit dem Cluster verknüpft ist, keine verfügbaren IP-Adressen hat. Eine AWS CloudHSM-Schlüsselspeicherverbindung erfordert eine freie IP-Adresse in jedem der zugehörigen privaten Subnetze, wobei es vorzugsweise zwei sind.

  Sie können in einem vorhandenen Subnetz keine IP-Adressen (CIDR-Blöcke) hinzufügen. Verschieben oder löschen Sie nach Möglichkeit andere Ressourcen, die die IP-Adressen im Subnetz verwenden, wie z. B. nicht verwendete EC2-Instances oder Elastic-Network-Schnittstellen. Andernfalls können Sie einen Cluster von einem aktuellen Backup des AWS CloudHSM-Clusters mit neuen oder vorhandenen privaten Subnetzen, die mehr freien Adressraum haben, erstellen. Wenn Sie dann den neuen Cluster mit Ihrem AWS CloudHSM-Schlüsselspeicher verknüpfen möchten, trennen Sie den benutzerdefinierten Schlüsselspeicher, ändern Sie die Cluster-ID des AWS CloudHSM-Schlüsselspeichers in die ID des neuen Clusters und versuchen Sie erneut, eine Verbindung herzustellen.

> **Tip**
>
> Um das Zurücksetzen des `kmsuser`-Passworts zu vermeiden, verwenden Sie die neueste Sicherung des AWS CloudHSM-Clusters.

- `INTERNAL_ERROR` gibt an, dass AWS KMS die Anforderung aufgrund eines internen Fehlers nicht ordnungsgemäß durchführen konnte. Wiederholen Sie die Anforderung. Trennen Sie bei `ConnectCustomKeyStore`-Anforderungen den AWS CloudHSM-Schlüsselspeicher, bevor Sie die Verbindung wieder herzustellen.
- `INVALID_CREDENTIALS` zeigt an, dass AWS KMS sich nicht beim zugehörigen AWS CloudHSM-Cluster anmelden kann, da es nicht über das korrekte `kmsuser`-Kontopasswort verfügt. Für Unterstützung bei diesem Fehler vgl. Reparieren ungültiger `kmsuser`-Anmeldeinformationen.
- `NETWORK_ERRORS` weist normalerweise auf vorübergehende Netzwerkprobleme hin. Trennen Sie den AWS CloudHSM-Schlüsselspeicher, warten Sie einige Minuten, und versuchen Sie dann erneut, eine Verbindung herzustellen.
- `SUBNET_NOT_FOUND` gibt an, dass mindestens ein Subnetz in der AWS CloudHSM-Clusterkonfiguration gelöscht wurde. Wenn AWS KMS nicht alle Subnetze in der Clusterkonfiguration finden kann, schlagen Versuche, den AWS CloudHSM-Schlüsselspeicher mit dem AWS CloudHSM-Cluster zu verbinden, fehl.

  Um diesen Fehler zu beheben, erstellen Sie einen Cluster aus einer möglichst neuen Sicherung desselben AWS CloudHSM-Clusters. (Dieser Prozess erstellt eine neue Clusterkonfiguration mit einer VPC und privaten Subnetzen.) Stellen Sie sicher, dass der neue Cluster die Anforderungen für einen benutzerdefinierten Schlüsselspeicher erfüllt, und notieren Sie sich die neue Cluster-ID. Wenn Sie dann den neuen Cluster mit Ihrem AWS CloudHSM-Schlüsselspeicher verknüpfen möchten, trennen Sie den benutzerdefinierten Schlüsselspeicher, ändern Sie die Cluster-ID des AWS CloudHSM-Schlüsselspeichers in die ID des neuen Clusters und versuchen Sie erneut, eine Verbindung herzustellen.

> **Tip**
>
> Um das Zurücksetzen des `kmsuser`-Passworts zu vermeiden, verwenden Sie die neueste Sicherung des AWS CloudHSM-Clusters.

- `USER_LOCKED_OUT` gibt an, dass das `kmsuser`-Kryptobenutzer (CU)-Konto aufgrund zu vieler fehlerhafter Passworteingaben aus dem zugehörigen AWS CloudHSM-Cluster ausgesperrt wurde. Für Unterstützung bei diesem Fehler vgl. Reparieren ungültiger `kmsuser`-Anmeldeinformationen.

  Um diesen Fehler zu beheben, trennen Sie den AWS CloudHSM-Schlüsselspeicher, und verwenden Sie den Befehl changePswd in cloudhsm_mgmt_util, um das `kmsuser`-Kontopasswort zu ändern. Bearbeiten Sie dann die `kmsuser`-Passworteinstellung für den Custom Key Store, und versuchen Sie erneut, eine Verbindung herzustellen. Falls Sie Hilfe benötigen, verwenden Sie die Vorgehensweise aus dem Thema Reparieren ungültiger `kmsuser`-Anmeldeinformationen.
- `USER_LOGGED_IN` gibt an, dass das `kmsuser`-CU-Konto beim zugehörigen AWS CloudHSM-Cluster angemeldet ist. Dadurch wird AWS KMS daran gehindert, das `kmsuser`-Kontopasswort zu rotieren und sich beim Cluster anzumelden. Melden Sie den `kmsuser`-CU vom Cluster ab, um diesen Fehler zu beheben. Wenn Sie das `kmsuser`-Kennwort für die Anmeldung beim Cluster geändert haben, müssen Sie auch den Passwortwert für den AWS CloudHSM-Schlüsselspeicher aktualisieren. Weitere Informationen dazu finden Sie unter Abmelden und erneutes Verbinden.
- `USER_NOT_FOUND` gibt an, dass AWS KMS kein `kmsuser`-CU-Konto im zugeordneten AWS CloudHSM-Cluster findet. Um diesen Fehler zu beheben, erstellen Sie ein `kmsuser` CU-Konto im Cluster, und aktualisieren Sie dann den Passwortwert für den AWS CloudHSM-Schlüsselspeicher. Weitere Informationen dazu finden Sie unter Reparieren ungültiger `kmsuser`-Anmeldeinformationen.

Wie man auf Fehler bei kryptografischen Produktionen reagiert

Eine kryptografische Produktion, die einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwendet, schlägt möglicherweise mit einem `KMSInvalidStateException` fehl. Die folgenden Fehlermeldungen könnten dem `KMSInvalidStateException` beiliegen.

KMS kann nicht mit Ihrem CloudHSM-Cluster kommunizieren. Dies könnte ein vorübergehendes Netzwerkproblem sein. Wenn dieser Fehler wiederholt auftritt, überprüfen Sie, ob die Netzwerk-ACLs und die Sicherheitsgruppenregeln für die VPC des AWS CloudHSM-Clusters korrekt sind.

- Obwohl dies ein HTTPS-400-Fehler ist, kann es auf vorübergehende Netzwerkprobleme zurückzuführen sein. Um zu antworten, versuchen Sie zunächst die Anforderung erneut. Wenn es jedoch weiterhin fehlschlägt, überprüfen Sie die Konfiguration der Netzwerkkomponenten. Dieser Fehler wird höchstwahrscheinlich durch die Fehlkonfiguration einer Netzwerkkomponente

verursacht, z. B. eine Firewall-Regel oder eine VPC -Sicherheitsgruppen-Regel, die ausgehenden Datenverkehr blockiert.

KMS kann nicht mit dem AWS CloudHSM-Cluster kommunizieren, da der kmsuser gesperrt ist. Wenn dieser Fehler wiederholt auftritt, trennen Sie die Verbindung zum AWS CloudHSM-Schlüsselspeicher und setzen Sie das Passwort für das kmsuser-Konto zurück. Aktualisieren Sie das kmsuser-Passwort für den benutzerdefinierten Schlüsselspeicher und versuchen Sie es erneut mit der Anfrage.

- Diese Fehlermeldung gibt an, dass das `kmsuser`-Crypto-Benutzer (CU)-Konto aufgrund zu vieler fehlerhafter Passworteingaben aus dem zugehörigen AWS CloudHSM-Cluster ausgesperrt wurde. Für Unterstützung bei diesem Fehler vgl. Trennen und Anmelden.

Reparieren ungültiger **`kmsuser`**-Anmeldeinformationen

Wenn Sie einen AWS CloudHSM-Schlüsselspeicher verbinden, meldet AWS KMS sich bei dem zugeordneten AWS CloudHSM-Cluster als `kmsuser`-Kryptobenutzer (Crypto User, CU) an. Die Anmeldung bleibt bestehen, bis der AWS CloudHSM-Schlüsselspeicher getrennt wird. Die Antwort DescribeCustomKeyStores meldet `ConnectionState` als `FAILED` und `ConnectionErrorCode` mit dem Wert `INVALID_CREDENTIALS` (siehe folgendes Beispiel).

Wenn Sie den AWS CloudHSM-Schlüsselspeicher trennen und das `kmsuser`-Passwort ändern, kann sich AWS KMS nicht beim AWS CloudHSM-Cluster mit den Anmeldeinformationen des `kmsuser`-CU-Kontos anmelden. Dies hat zur Folge, dass alle Versuche, den AWS CloudHSM-Schlüsselspeicher zu verbinden, fehlschlagen. Die Antwort `DescribeCustomKeyStores` meldet `ConnectionState` als `FAILED` und `ConnectionErrorCode` mit dem Wert `INVALID_CREDENTIALS` (siehe folgendes Beispiel).

```
$ 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"
   ],
}
```

Weiterhin gilt, dass, AWS CloudHSM nach fünf fehlgeschlagenen Anmeldeversuchen bei dem Cluster mit einem inkorrekten Passwort das Benutzerkonto sperrt. Zur Anmeldung bei dem Cluster müssen Sie das Kontopasswort ändern.

Wenn AWS KMS beim Versuch zur Anmeldung bei dem Cluster als `kmsuser`-CU eine Lockout-Antwort erhält, schlägt der Verbindungsversuch für den AWS CloudHSM-Schlüsselspeicher fehl. Die DescribeCustomKeyStores Antwort enthält einen `ConnectionState` von `FAILED` und einen `ConnectionErrorCode` Wert von `USER_LOCKED_OUT`, wie im folgenden Beispiel gezeigt.

```
$ 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"
   ],
}
```

Gehen Sie zum Beheben eines dieser Zustände wie folgt vor.

1. Trennen Sie den AWS CloudHSM-Schlüsselspeicher.
2. Führen Sie die -DescribeCustomKeyStoresOperation aus und zeigen Sie den Wert des -`ConnectionErrorCode`Elements in der Antwort an.

   - Wenn der `ConnectionErrorCode`-Wert `INVALID_CREDENTIALS` ist, ermitteln Sie das aktuelle Passwort für das `kmsuser`-Konto. Verwenden Sie bei Bedarf den Befehl changePswd in cloudhsm_mgmt_util, um das Passwort auf einen bekannten Wert zu setzen.
   - Wenn der `ConnectionErrorCode`-Wert `USER_LOCKED_OUT` ist, müssen Sie den Befehl changePswd in cloudhsm_mgmt_util verwenden, um das `kmsuser`-Passwort zu ändern.
3. Bearbeiten Sie die `kmsuser` Passworteinstellung, so dass sie dem aktuellen `kmsuser`-Passwort in dem Cluster entspricht. Diese Aktion weist AWS KMS an, welches Passwort für die

Anmeldung bei dem Cluster zu verwenden ist. Das `kmsuser`-Passwort in dem Cluster wird nicht geändert.

4. Verbinden Sie den Custom Key Store.

Löschen von verwaistem Schlüsselmaterial

Nach der Planung der Löschung eines KMS-Schlüssels aus einem AWS CloudHSM-Schlüsselspeicher müssen Sie möglicherweise das entsprechende Schlüsselmaterial manuell aus dem zugehörigen AWS CloudHSM-Cluster löschen.

Wenn Sie einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher erstellen, erstellt AWS KMS die KMS-Schlüssel-Metadaten in AWS KMS und generiert das Schlüsselmaterial in dem zugeordneten AWS CloudHSM-Cluster. Wenn Sie die Löschung eines KMS-Schlüssels in einem AWS CloudHSM-Schlüsselspeicher planen, löscht AWS KMS nach der Wartezeit die KMS-Schlüssel-Metadaten. Anschließend versucht AWS KMS unter Verwendung aller verfügbaren Methoden, das Schlüsselmaterial aus dem zugehörigen AWS CloudHSM-Cluster zu löschen. Der Versuch kann fehlschlagen, wenn AWS KMS nicht auf den Cluster zugreifen kann, z. B. wenn die Verbindung zum AWS CloudHSM-Schlüsselspeicher getrennt wurde oder sich das `kmsuser` Kennwort geändert hat. AWS KMS versucht nicht, das Material des Schlüssels aus Clustersicherungen zu löschen.

AWS KMS meldet die Ergebnisse des Versuchs, das Schlüsselmaterial aus dem Cluster im `DeleteKey`-Eventeintrag Ihrer AWS CloudTrail-Protokolle zu löschen. Es erscheint im `backingKeysDeletionStatus`-Element des `additionalEventData`-Elements, wie im folgenden Beispieleintrag gezeigt. Der Eintrag enthält die KMS-Schlüssel-ID, die AWS CloudHSM-Cluster-ID und den Schlüssel-Handle des Schlüsselmaterials (`backing-key-id`).

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
```

```
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"16\",\"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"
}
```

Gehen Sie zum Löschen des Schlüsselmaterials von dem zugehörigen AWS CloudHSM-Cluster etwa wie folgt vor. Dieses Beispiel verwendet die Befehlszeilen-Tools AWS CLI und AWS CloudHSM, Sie können stattdessen aber auch die AWS Management Console verwenden.

1. Trennen Sie, falls noch nicht geschehen, den AWS CloudHSM-Schlüsselspeicher, und melden Sie sich dann bei key_mgmt_util an, wie in Trennen und Anmelden erläutert.
2. Verwenden Sie den Befehl deleteKey in key_mgmt_util, um den Schlüssel von den HSMs in dem Cluster zu löschen.

   Beispielsweise löscht dieser Befehl den Schlüssel 262162 von den HSMs in dem Cluster. Das Schlüssel-Handle ist im CloudTrail Protokolleintrag aufgeführt.

   ```
   Command: deleteKey -k 262162

           Cfm3DeleteKey returned: 0x00 : HSM Return: SUCCESS

           Cluster Error Status
           Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
           Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
   ```

```
        Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

3. Melden Sie sich von key_mgmt_util ab, und verbinden Sie den AWS CloudHSM-Schlüsselspeicher wieder wie in Abmelden und erneutes Verbinden beschrieben.

Wiederherstellen von gelöschtem Schlüsselmaterial für einen KMS-Schlüssel

Wenn das Schlüsselmaterial für einen AWS KMS key gelöscht wird, kann der KMS-Schlüssel nicht verwendet werden und der gesamte Chiffretext, der unter dem KMS-Schlüssel verschlüsselt wurde, kann nicht entschlüsselt werden. Dies kann der Fall sein, wenn das Schlüsselmaterial für einen KMS-Schlüssel in einem AWS CloudHSM-Schlüsselspeicher aus dem zugeordneten AWS CloudHSM-Cluster gelöscht wurde. Es kann jedoch möglich sein, das Schlüsselmaterial wiederherzustellen.

Wenn Sie einen AWS KMS key (KMS-Schlüssel) in einem AWS CloudHSM-Schlüsselspeicher erstellen, meldet sich AWS KMS im zugeordneten AWS CloudHSM-Cluster an und erstellt das Schlüsselmaterial für den KMS-Schlüssel. Dazu ändert es das Passwort zu einem Wert, der nur ihm bekannt ist, und bleibt angemeldet, bis der AWS CloudHSM-Schlüsselspeicher verbunden wird. Da nur der Eigentümer des Schlüssels, d.h. der CU, der den Schlüssel erstellt hat, diesen löschen kann, ist es unwahrscheinlich, dass der Schlüssel versehentlich von den HSMs gelöscht wird.

Wenn jedoch das Schlüsselmaterial für einen KMS-Schlüssel von den HSMs in einem Cluster gelöscht wird, wechselt der Schlüsselstatus des KMS-Schlüssels schließlich zu `UNAVAILABLE`. Wenn Sie versuchen, den KMS-Schlüssel für eine kryptographische Operation zu verwenden, schlägt die Operation mit einer `KMSInvalidStateException`-Ausnahme fehl. Wichtig dabei ist, dass die mit dem KMS-Schlüssel verschlüsselten Daten nicht mehr entschlüsselt werden können.

Unter bestimmten Umständen können Sie das gelöschte Schlüsselmaterial wiederherstellen, indem Sie einen Cluster aus einem Backup erstellen, das das Schlüsselmaterial enthält. Diese Strategie funktioniert nur, wenn mindestens eine Sicherung erstellt wurde, während der Schlüssel vorhanden war und bevor er gelöscht wurde.

Gehen Sie wie folgt vor, um das Schlüsselmaterial wiederherzustellen.

1. Suchen Sie ein Cluster-Backup, das das Schlüsselmaterial enthält. Die Sicherung muss dazu alle Benutzer und Schlüssel enthalten, die zur Unterstützung des Clusters und seiner verschlüsselten Daten erforderlich sind.

   Verwenden Sie die -DescribeBackupsOperation, um die Backups für einen Cluster aufzulisten. Verwenden Sie dann den Sicherung-Zeitstempel, um eine Sicherung auszuwählen. Um die

Ausgabe auf den Cluster zu beschränken, der mit dem AWS CloudHSM-Schlüsselspeicher verbunden ist, verwenden Sie den Parameter `Filters`, wie im folgenden Beispiel gezeigt.

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
{
    "Backups": [
        {
            "ClusterId": "cluster-1a23b4cdefg",
            "BackupId": "backup-9g87f6edcba",
            "CreateTimestamp": 1536667238.328,
            "BackupState": "READY"
        },
             ...
    ]
}
```

2. Erstellen Sie einen Cluster aus der ausgewählten Sicherung. Prüfen Sie, ob die Sicherung den gelöschten Schlüssel sowie weitere Benutzer und Schlüssel enthält, die für den Cluster erforderlich sind.
3. Trennen Sie den AWS CloudHSM-Schlüsselspeicher, damit Sie seine Eigenschaften bearbeiten können.
4. Bearbeiten Sie die Cluster-ID des AWS CloudHSM-Schlüsselspeichers. Geben Sie die Cluster-ID des Clusters ein, den Sie aus der Sicherung erstellt haben. Da der Cluster seinen Sicherungsverlauf mit dem ursprünglichen Cluster gemeinsam hat, sollte die neue Cluster-ID gültig sein.
5. Verbinden Sie den AWS CloudHSM-Schlüsselspeicher erneut.

Anmeldung als **`kmsuser`**

Um Schlüsselmaterial in dem AWS CloudHSM-Cluster für Ihren AWS CloudHSM-Schlüsselspeicher zu erstellen und zu verwalten, verwendet AWS KMS das `kmsuser`-Kryptobenutzer-Konto (CU). Sie erstellen das `kmsuser`-CU-Konto in Ihrem Cluster und übergeben dessen Passwort an AWS KMS, wenn Sie Ihren AWS CloudHSM-Schlüsselspeicher erstellen.

Generell verwaltet AWS KMS das `kmsuser`-Konto. Für einige Aufgaben müssen Sie jedoch den AWS CloudHSM-Schlüsselspeicher trennen, sich bei dem Cluster als `kmsuser`-CU anmelden und die Befehlszeilen-Tools cloudhsm_mgmt_util und key_mgmt_util verwenden.

> **Note**
>
> Sämtliche Versuche, KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen oder vorhandene KMS-Schlüssel in kryptografischen Produktionen zu nutzen, schlagen fehl, während der benutzerdefinierte Schlüsselspeicher getrennt ist. Diese Aktion kann verhindern, dass Benutzer vertrauliche Daten speichern und darauf zugreifen.

In diesem Thema wird erläutert, wie Sie Ihren AWS CloudHSM-Schlüsselspeicher trennen und sich als `kmsuser` anmelden, das AWS CloudHSM-Befehlszeilen-Tool ausführen und sich abmelden und Ihren AWS CloudHSM-Schlüsselspeicher wieder verbinden.

Themen

- Trennen und Anmelden
- Abmelden und erneutes Verbinden

Trennen und Anmelden

Gehen Sie jedes Mal wie folgt vor, wenn Sie sich bei einem zugehörigen Cluster als `kmsuser`-CU anmelden müssen.

1. Trennen Sie den AWS CloudHSM-Schlüsselspeicher, falls noch nicht geschehen. Sie können die AWS KMS-Konsole oder die AWS KMS-API verwenden.

   Während Ihrer AWS CloudHSM Schlüssel verbunden ist, wird AWS KMS als `kmsuser` angemeldet. Dadurch wird verhindert, dass Sie sich als `kmsuser` anmelden oder das `kmsuser`-Passwort ändern.

   Dieser Befehl verwendet beispielsweise , DisconnectCustomKeyStore um einen Beispiel-Schlüsselspeicher zu trennen. Ersetzen Sie die Beispiel-ID des AWS CloudHSM-Schlüsselspeichers durch eine gültige ID.

   ```
   $ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

2. Starten Sie cloudhsm_mgmt_util. Verwenden Sie das im Abschnitt Vorbereitung zur Ausführung von cloudhsm_mgmt_util des AWS CloudHSM-Benutzerhandbuch beschriebene Verfahren.
3. Melden Sie sich bei cloudhsm_mgmt_util auf dem AWS CloudHSM-Cluster als Verschlüsselungsverantwortlicher (Crypto Officer, CO) an.

Dieser Befehl führt beispielsweise zur Anmeldung eines COs namens „admin“. Ersetzen Sie den Beispiel-CO-Benutzernamen und das Passwort durch gültige Werte.

```
aws-cloudhsm>loginHSM CO admin <password>
loginHSM success on server 0(10.0.2.9)
loginHSM success on server 1(10.0.3.11)
loginHSM success on server 2(10.0.1.12)
```

4. Verwenden Sie den Befehl changePswd, um das Passwort des `kmsuser`-Kontos zu einem Passwort zu ändern, das Sie kennen. (AWS KMS rotiert das Passwort, wenn Sie Ihren AWS CloudHSM-Schlüsselspeicher verbinden.) Das Passwort muss 7 – 32 alphanumerische Zeichen umfassen. Bei der Angabe wird zwischen Groß- und Kleinschreibung unterschieden, Sonderzeichen sind nicht zulässig.

   Beispielsweise ändert dieser Befehl das `kmsuser`-Passwort zu `tempPassword`.

   ```
   aws-cloudhsm>changePswd CU kmsuser tempPassword

   ************************CAUTION*********************************
   This is a CRITICAL operation, should be done on all nodes in the
   cluster. Cav server does NOT synchronize these changes with the
   nodes on which this operation is not executed or failed, please
   ensure this operation is executed on all nodes in the cluster.
   ****************************************************************

   Do you want to continue(y/n)?y
   Changing password for kmsuser(CU) on 3 nodes
   ```

5. Melden Sie sich bei key_mgmt_util oder cloudhsm_mgmt_util als `kmsuser` mit dem von ihnen eingerichteten Passwort an. Detaillierte Anleitungen finden Sie unter Erste Schritte mit cloudhsm_mgmt_util und Erste Schritte mit key_mgmt_util. Das Tool, das Sie verwenden, hängt von Ihrer Aufgabe ab.

   Beispielsweise führt dieser Befehl zur Anmeldung bei key_mgmt_util.

   ```
   Command: loginHSM -u CU -s kmsuser -p tempPassword
   Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

   Cluster Error Status
   Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
   Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
   ```

```
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

Abmelden und erneutes Verbinden

1. Führen Sie die Aufgabe durch, und melden Sie sich dann von dem Befehlszeilen-Tool ab. Wenn Sie sich nicht abmelden, schlagen Versuche, Ihren AWS CloudHSM-Schlüsselspeicher wieder zu verbinden, fehl.

   ```
   Command:  logoutHSM
   Cfm3LogoutHSM returned: 0x00 : HSM Return: SUCCESS

   Cluster Error Status
   Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
   Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
   ```

2. Bearbeiten Sie die `kmsuser`-Passworteinstellung für den Custom Key Store.

   Dadurch wird AWS KMS das aktuelle Passwort für `kmsuser` in dem Cluster mitgeteilt. Wenn Sie diesen Schritt auslassen, kann sich AWS KMS nicht bei dem Cluster als `kmsuser` anmelden, und alle Versuche, ihren Custom Key Store wieder zu verbinden, schlagen fehl. Sie können die -AWS KMSKonsole oder den -`KeyStorePassword`Parameter der -UpdateCustomKeyStoreOperation verwenden.

   Beispielsweise teilt dieser Befehl AWS KMS mit, dass das aktuelle Passwort `tempPassword` ist. Ersetzen Sie das Beispielpasswort durch das tatsächliche Passwort.

   ```
   $ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
   key-store-password tempPassword
   ```

3. Verbinden Sie den AWS KMS-Schlüsselspeicher wieder mit dessen AWS CloudHSM-Cluster. Ersetzen Sie die Beispiel-ID des AWS CloudHSM-Schlüsselspeichers durch eine gültige ID. Während des Verbindungsvorgangs ändert AWS KMS das `kmsuser`-Passwort zu einem Wert, der nur ihm bekannt ist.

   Der ConnectCustomKeyStore Vorgang kehrt schnell zurück, aber der Verbindungsprozess kann einen längeren Zeitraum in Anspruch nehmen. Die erste Reaktion ist kein Zeichen für den Erfolg des Verbindungsvorgangs.

   ```
   $ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

4. Verwenden Sie die -DescribeCustomKeyStoresOperation, um zu überprüfen, ob der -AWS CloudHSMSchlüsselspeicher verbunden ist. Ersetzen Sie die Beispiel-ID des AWS CloudHSM-Schlüsselspeichers durch eine gültige ID.

   In diesem Beispiel zeigt das Verbindungsfeld, dass der AWS CloudHSM-Schlüsselspeicher jetzt verbunden ist.

```
$ 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"
   ],
}
```

Externe Schlüsselspeicher

Externe Schlüsselspeicher ermöglichen es Ihnen, Ihre AWS Ressourcen mithilfe kryptografischer Schlüssel außerhalb von AWS zu schützen. Dieses erweiterte Feature wurde für regulierte Workloads entwickelt, die Sie mit Verschlüsselungsschlüsseln schützen müssen, die in einem externen System für die Schlüsselverwaltung gespeichert sind, das Sie kontrollieren. Externe Schlüsselspeicher unterstützen das Versprechen der AWS digitalen Souveränität und geben Ihnen die souveräne Kontrolle über Ihre Daten AWS, einschließlich der Möglichkeit, mit Schlüsselmaterial zu verschlüsseln, das Ihnen gehört und das Sie außerhalb kontrollieren. AWS

Ein externer Schlüsselspeicher ist ein benutzerdefinierter Schlüsselspeicher, der von einem externen Schlüsselmanager unterstützt wird, den Sie besitzen und den Sie außerhalb verwalten. AWS Ihr externer Schlüsselmanager kann ein physisches oder virtuelles Hardwaresicherheitsmodul (HSM) oder ein beliebiges hardware- oder softwarebasiertes System sein, das kryptografische Schlüssel generieren und verwenden kann. Verschlüsselungs- und Entschlüsselungsvorgänge, bei denen ein KMS-Schlüssel in einem externen Schlüsselspeicher verwendet wird, werden von Ihrem externen Schlüsselmanager unter Verwendung Ihres kryptografischen Schlüsselmaterials ausgeführt. Dieses Feature wird als Hold Your Own Keys (HYOKs) bezeichnet.

AWS KMS interagiert nie direkt mit Ihrem externen Schlüsselmanager und kann Ihre Schlüssel nicht erstellen, anzeigen, verwalten oder löschen. AWS KMS Interagiert stattdessen nur mit der externen Schlüsselspeicher-Proxy-Software (XKS-Proxy), die Sie bereitstellen. Ihr externer Schlüsselspeicher-Proxy vermittelt die gesamte Kommunikation zwischen AWS KMS und Ihrem externen Schlüsselmanager. Er überträgt alle Anfragen von AWS KMS Ihrem externen Schlüsselmanager und überträgt die Antworten von Ihrem externen Schlüsselmanager zurück an AWS KMS. Der externe Schlüsselspeicher-Proxy übersetzt auch generische Anfragen AWS KMS in ein herstellerspezifisches Format, das Ihr externer Schlüsselmanager verstehen kann, sodass Sie externe Schlüsselspeicher mit Schlüsselmanagern verschiedener Anbieter verwenden können.

Sie können KMS-Schlüssel in einem externen Schlüsselspeicher für die clientseitige Verschlüsselung verwenden, auch mit dem AWS Encryption SDK. Externe Schlüsselspeicher sind jedoch eine wichtige Ressource für die serverseitige Verschlüsselung, sodass Sie Ihre AWS Ressourcen mehrfach AWS-Services mit Ihren kryptografischen Schlüsseln außerhalb von schützen können. AWS AWS-Services die vom Kunden verwaltete Schlüssel für symmetrische Verschlüsselung unterstützen, unterstützen auch KMS-Schlüssel in einem externen Schlüsselspeicher. Einzelheiten zum Servicesupport finden Sie unter AWS -Serviceintegration.

Externe Schlüsselspeicher ermöglichen die Verwendung AWS KMS für regulierte Workloads, bei denen Verschlüsselungsschlüssel außerhalb von gespeichert und verwendet werden müssen. AWS Sie weichen jedoch erheblich vom Standardmodell der übergreifenden Verantwortlichkeit ab und sind mit zusätzlichen operativen Belastungen verbunden. Das höhere Risiko für Verfügbarkeit und Latenz wird für die meisten Kunden die wahrgenommenen Sicherheitsvorteile externer Schlüsselspeicher übersteigen.

Mit externen Schlüsselspeichern haben Sie die Kontrolle über den Vertrauensanker. Daten, die mit KMS-Schlüsseln in Ihrem externen Schlüsselspeicher verschlüsselt wurden, können nur mithilfe des von Ihnen kontrollierten externen Schlüsselmanagers entschlüsselt werden. Wenn Sie den Zugriff auf Ihren externen Schlüsselmanager vorübergehend entziehen, z. B. indem Sie die Verbindung zum externen Schlüsselspeicher trennen oder Ihren externen Schlüsselmanager vom externen Schlüsselspeicher-Proxy trennen, AWS geht jeglicher Zugriff auf Ihre kryptografischen Schlüssel verloren, bis Sie ihn wiederherstellen. Während dieses Intervalls kann der unter Ihren KMS-Schlüsseln verschlüsselte Geheimtext nicht entschlüsselt werden. Wenn Sie den Zugriff auf Ihren externen Schlüsselmanager dauerhaft widerrufen, kann der gesamte unter einem KMS-Schlüssel verschlüsselte Geheimtext in Ihrem externen Schlüsselspeicher nicht wiederhergestellt werden. Die einzigen Ausnahmen sind AWS Dienste, die die durch Ihre KMS-Schlüssel geschützten Datenschlüssel kurzzeitig zwischenspeichern. Diese Datenschlüssel funktionieren so lange, bis Sie

die Ressource deaktivieren oder der Cache abläuft. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Externe Schlüsselspeicher schließen die wenigen Anwendungsfälle für regulierte Workloads aus, bei denen Verschlüsselungsschlüssel ausschließlich unter Ihrer Kontrolle bleiben müssen und für die Sie keinen Zugriff haben müssen. AWS Dies ist jedoch eine große Änderung in der Art und Weise, wie Sie eine cloudbasierte Infrastruktur betreiben, und eine bedeutende Änderung des Modells der übergreifenden Verantwortlichkeit. Bei den meisten Workloads werden der zusätzliche Betriebsaufwand und die höheren Risiken für Verfügbarkeit und Leistung die wahrgenommenen Sicherheitsvorteile externer Schlüsselspeicher übersteigen.

Weitere Informationen:

- Ankündigung des AWS KMS External Key Store im AWS News-Blog.

Benötige ich einen externen Schlüsselspeicher?

Für die meisten Benutzer erfüllt der AWS KMS Standard-Schlüsselspeicher, der durch FIPS 140-2 Security Level 3-validierte Hardware-Sicherheitsmodule geschützt ist, ihre Sicherheits-, Kontroll- und behördlichen Anforderungen. Externe Benutzer von Schlüsselspeichern haben einen erheblichen Aufwand an Kosten, Wartung und Fehlerbehebung sowie Risiken in Bezug auf Latenz, Verfügbarkeit und Zuverlässigkeit.

Wenn Sie einen externen Schlüsselspeicher in Betracht ziehen, sollten Sie sich etwas Zeit nehmen, um die Alternativen zu verstehen. Dazu gehören ein AWS CloudHSM -Schlüsselspeicher, der von einem AWS CloudHSM -Cluster unterstützt wird, den Sie besitzen und verwalten, und KMS-Schlüssel mit importiertem Schlüsselmaterial, die Sie in Ihren eigenen HSMs generieren und die Sie bei Bedarf aus den KMS-Schlüsseln löschen können. Insbesondere der Import von Schlüsselmaterial mit einem sehr kurzen Verfallsintervall könnte ein ähnliches Maß an Kontrolle bieten, ohne die Leistungs- oder Verfügbarkeitsrisiken einzugehen.

Ein externer Schlüsselspeicher könnte die richtige Lösung für Ihr Unternehmen sein, wenn Sie die folgenden Anforderungen erfüllen:

- Sie müssen kryptografische Schlüssel in Ihrem lokalen Schlüsselmanager oder einem Schlüsselmanager verwenden, den Sie nicht kontrollieren. AWS

- Sie müssen nachweisen, dass Ihre kryptografischen Schlüssel außerhalb der Cloud ausschließlich unter Ihrer Kontrolle aufbewahrt werden.
- Sie müssen mithilfe kryptografischer Schlüssel mit unabhängiger Autorisierung verschlüsseln und entschlüsseln.
- Für Schlüsselmaterial muss ein unabhängiger sekundärer Prüfungspfad eingerichtet sein.

Wenn Sie sich für einen externen Schlüsselspeicher entscheiden, beschränken Sie dessen Verwendung auf Workloads, die außerhalb von AWS mit kryptografischen Schlüsseln geschützt werden müssen.

Modell der geteilten Verantwortung

Standardmäßige KMS-Schlüssel verwenden Schlüsselmaterial, das in HSMs generiert und verwendet wird, die AWS KMS Eigentümer und Verwalter sind. Sie legen die Zugriffskontrollrichtlinien für Ihre KMS-Schlüssel fest und konfigurieren AWS-Services , dass KMS-Schlüssel zum Schutz Ihrer Ressourcen verwendet werden. AWS KMS übernimmt die Verantwortung für die Sicherheit, Verfügbarkeit, Latenz und Haltbarkeit des Schlüsselmaterials in Ihren KMS-Schlüsseln.

KMS-Schlüssel in externen Schlüsselspeichern hängen von wichtigen Materialien und Vorgängen in Ihrem externen Schlüsselmanager ab. Somit verschiebt sich das Gleichgewicht der Verantwortung in Ihre Richtung. Sie sind für die Sicherheit, Zuverlässigkeit, Haltbarkeit und Leistung der kryptografischen Schlüssel in Ihrem externen Schlüsselmanager verantwortlich. AWS KMS ist dafür verantwortlich, umgehend auf Anfragen zu antworten und mit Ihrem externen Schlüsselspeicher-Proxy zu kommunizieren und unsere Sicherheitsstandards aufrechtzuerhalten. Um sicherzustellen, dass jeder externe Schlüssel den Chiffretext mindestens so stark speichert wie der AWS KMS Standard-Chiffretext, verschlüsselt er AWS KMS zunächst den gesamten Klartext mit AWS KMS Schlüsselmaterial, das für Ihren KMS-Schlüssel spezifisch ist, und sendet ihn dann zur Verschlüsselung mit Ihrem externen Schlüssel an Ihren externen Schlüsselmanager. Dieses Verfahren wird als doppelte Verschlüsselung bezeichnet. Infolgedessen können weder AWS KMS noch der/die Besitzer:in des externen Schlüsselmaterials doppelt verschlüsselten Geheimtext alleine entschlüsseln.

Sie sind verantwortlich für die Pflege eines externen Schlüsselmanagers, der Ihren gesetzlichen und Leistungsstandards entspricht, für die Bereitstellung und Pflege eines externen Schlüsselspeicher-Proxys, der der API-Spezifikation für externe Schlüsselspeicher-Proxys von AWS KMS entspricht, und für die Gewährleistung der Verfügbarkeit und Beständigkeit Ihres Schlüsselmaterials. Sie

müssen auch einen externen Schlüsselspeicher erstellen, konfigurieren und verwalten. Wenn Fehler auftreten, die durch Komponenten verursacht werden, die Sie warten, müssen Sie darauf vorbereitet sein, die Fehler zu identifizieren und zu beheben, damit die AWS Dienste ohne übermäßige Unterbrechung auf Ihre Ressourcen zugreifen können. AWS KMS bietet Anleitungen zur Problembehebung, mit deren Hilfe Sie die Ursache von Problemen und die wahrscheinlichsten Lösungen ermitteln können.

Überprüfen Sie die CloudWatch Amazon-Metriken und -Dimensionen, die für externe Schlüsselgeschäfte AWS KMS aufgezeichnet wurden. AWS KMS empfiehlt dringend, CloudWatch Alarme zur Überwachung Ihres externen Schlüsselspeichers einzurichten, damit Sie die ersten Anzeichen von Leistungs- und Betriebsproblemen erkennen können, bevor sie auftreten.

Was ändert sich?

Externe Schlüsselspeicher unterstützen nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Innerhalb AWS KMS verwenden und verwalten Sie KMS-Schlüssel in einem externen Schlüsselspeicher auf die gleiche Weise wie andere vom Kunden verwaltete Schlüssel, einschließlich der Festlegung von Zugriffskontrollrichtlinien und der Überwachung der Schlüsselverwendung. Sie verwenden dieselben APIs mit denselben Parametern, um eine kryptografische Operation mit einem KMS-Schlüssel in einem externen Schlüsselspeicher anzufordern, den Sie für einen beliebigen KMS-Schlüssel verwenden. Der Preis ist derselbe wie für die Standard-KMS-Schlüssel. Weitere Informationen finden Sie unter Verwalten von KMS-Schlüsseln in einem externen Schlüsselspeicher, Verwenden von KMS-Schlüsseln in einem externen Schlüsselspeicher und AWS Key Management Service -Preise.

Bei externen Schlüsselspeichern ändern sich jedoch die folgenden Prinzipien:

- Sie sind für die Verfügbarkeit, Beständigkeit und Latenz wichtiger Vorgänge verantwortlich.
- Sie sind für alle Kosten für die Entwicklung, den Erwerb, den Betrieb und die Lizenzierung Ihres externen Schlüsselmanagersystems verantwortlich.
- Sie können die unabhängige Autorisierung aller Anfragen AWS KMS an Ihren externen Schlüsselspeicher-Proxy implementieren.
- Sie können alle Vorgänge Ihres externen Schlüsselspeicher-Proxys sowie alle Vorgänge Ihres externen Schlüsselmanagers im Zusammenhang mit AWS KMS Anfragen überwachen, prüfen und protokollieren.

Wo fange ich an?

Um einen externen Schlüsselspeicher zu erstellen und zu verwalten, müssen Sie die Proxy-Konnektivitätsoption für Ihren externen Schlüsselspeicher auswählen, die Voraussetzungen erfüllen und Ihren externen Schlüsselspeicher erstellen und konfigurieren. Um damit zu beginnen, siehe Planen eines externen Schlüsselspeichers.

Kontingente

AWS KMS ermöglicht bis zu 10 benutzerdefinierte Schlüsselspeicher in jeder AWS-Konto Region, einschließlich AWS CloudHSM Schlüsselspeichern und externer Schlüsselspeicher, unabhängig von deren Verbindungsstatus. Darüber hinaus gibt es AWS KMS -Anforderungskontingente für die Nutzung von KMS-Schlüsseln in einem externen Schlüsselspeicher.

Wenn Sie die VPC-Proxy-Konnektivität für Ihren externen Schlüsselspeicher-Proxy wählen, gelten möglicherweise auch Kontingente für die erforderlichen Komponenten wie VPCs, Subnetze und Network Load Balancer. Informationen über diese Kontingente erhalten Sie in der Service Quotas-Konsole.

Regionen

Um die Netzwerklatenz zu minimieren, erstellen Sie Ihre externen Schlüsselspeicherkomponenten in der AWS-Region , die Ihrem externen Schlüsselmanager am nächsten liegt. Wählen Sie nach Möglichkeit eine Region mit einer Netzwerk-Round-Trip-Zeit (RTT) von maximal 35 Millisekunden.

Externe Schlüsselspeicher werden AWS-Regionen in allen Bereichen unterstützt, AWS KMS mit Ausnahme von China (Peking) und China (Ningxia).

Nicht unterstützte Funktionen

AWS KMS unterstützt die folgenden Funktionen in benutzerdefinierten Schlüsselspeichern nicht.

- Asymmetrische KMS-Schlüssel
- Asymmetrische Datenschlüsselpaare
- HMAC-KMS-Schlüssel
- KMS-Schlüssel mit importiertem Schlüsselmaterial
- Automatische Schlüsselrotation
- Multiregionale Schlüssel

Themen

Konzepte für externe Schlüsselspeicher

In diesem Thema erläutern wir einige der Konzepte in externen Schlüsselspeichern.

Themen

Externer Schlüsselspeicher

Ein externer Schlüsselspeicher ist ein AWS KMS benutzerdefinierter Schlüsselspeicher, der von einem externen Schlüsselmanager unterstützt wird AWS , der nicht Ihnen gehört und von dem Sie verwaltet werden. Jeder KMS-Schlüssel in einem externen Schlüsselspeicher ist mit einem externen Schlüssel in Ihrem externen Schlüsselmanager verknüpft. Wenn Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher für die Verschlüsselung oder Entschlüsselung verwenden, wird der Vorgang in Ihrem externen Schlüsselmanager unter Verwendung Ihres externen Schlüssels ausgeführt. Diese Anordnung wird als Hold Your Own Keys (HYOK) bezeichnet. Dieses Feature wurde für Organisationen entwickelt, die ihre kryptografischen Schlüssel in ihrem eigenen externen Schlüsselmanager verwalten müssen.

Externe Schlüsselspeicher stellen sicher, dass die kryptografischen Schlüssel und Operationen, die Ihre AWS Ressourcen schützen, in Ihrem externen Schlüsselmanager unter Ihrer Kontrolle verbleiben. AWS KMS sendet Anfragen an Ihren externen Schlüsselmanager zum Verschlüsseln und Entschlüsseln von Daten, AWS KMS kann jedoch keine externen Schlüssel erstellen, löschen oder verwalten. Alle Anfragen AWS KMS an Ihren externen Schlüsselmanager werden über eine externe Schlüsselspeicher-Proxy-Softwarekomponente vermittelt, die Sie bereitstellen, besitzen und verwalten.

AWS Dienste, die vom AWS KMS Kunden verwaltete Schlüssel unterstützen, können die KMS-Schlüssel in Ihrem externen Schlüsselspeicher verwenden, um Ihre Daten zu schützen. Infolgedessen werden Ihre Daten letztendlich durch Ihre Schlüssel geschützt, indem Sie Ihre Verschlüsselungsvorgänge in Ihrem externen Schlüsselmanager verwenden.

Für die KMS-Schlüssel in einem externen Schlüsselspeicher gelten grundsätzlich andere Vertrauensmodelle, Regelungen zur gemeinsamen Verantwortung und Leistungserwartungen als für Standard-KMS-Schlüssel. Bei externen Schlüsselspeichern sind Sie für die Sicherheit und Integrität des Schlüsselmaterials und der kryptografischen Operationen verantwortlich. Die Verfügbarkeit und Latenz von KMS-Schlüsseln in einem externen Schlüsselspeicher werden durch die Hardware, Software, Netzwerkkomponenten und die Entfernung zwischen AWS KMS und Ihrem externen Schlüsselmanager beeinflusst. Außerdem werden Ihnen wahrscheinlich zusätzliche Kosten für Ihren externen Schlüsselmanager und für die Netzwerk- und Lastausgleichsinfrastruktur entstehen, mit der Ihr externer Schlüsselmanager kommunizieren muss AWS KMS

Sie können Ihren externen Schlüsselspeicher als Teil Ihrer umfassenderen Datenschutzstrategie verwenden. Für jede AWS Ressource, die Sie schützen, können Sie entscheiden, welche einen KMS-Schlüssel in einem externen Schlüsselspeicher erfordern und welche durch einen Standard-KMS-Schlüssel geschützt werden können. Dies gibt Ihnen die Flexibilität, KMS-Schlüssel für bestimmte Datenklassifizierungen, Anwendungen oder Projekte auszuwählen.

Externer Schlüsselmanager

Ein externer Schlüsselmanager ist eine Komponente außerhalb von AWS , die symmetrische 256-Bit-AES-Schlüssel generieren und eine symmetrische Verschlüsselung und Entschlüsselung durchführen kann. Der externe Schlüsselmanager für einen externen Schlüsselspeicher kann ein physisches Hardwaresicherheitsmodul (HSM), ein virtuelles HSM oder ein Software-Schlüsselmanager mit oder ohne HSM-Komponente sein. Es kann sich an einem beliebigen Ort außerhalb von befinden AWS, auch bei Ihnen vor Ort, in einem lokalen oder entfernten Rechenzentrum oder in einer beliebigen Cloud. Ihr externer Schlüsselspeicher kann von einem einzelnen externen Schlüsselmanager oder mehreren zugehörigen Schlüsselmanager-Instances unterstützt werden, die kryptografische

Schlüssel gemeinsam nutzen, z. B. einen HSM-Cluster. Externe Schlüsselspeicher sind so konzipiert, dass sie eine Vielzahl von externen Managern verschiedener Anbieter unterstützen. Einzelheiten zu den Anforderungen an Ihren externen Schlüsselmanager finden Sie unter Planen eines externen Schlüsselspeichers.

Externer Schlüssel

Jeder KMS-Schlüssel in einem externen Schlüsselspeicher ist mit einem kryptografischen Schlüssel in Ihrem externen Schlüsselmanager verknüpft, der als externer Schlüssel bezeichnet wird. Wenn Sie mit einem KMS-Schlüssel in Ihrem externen Schlüsselspeicher verschlüsseln oder entschlüsseln, wird die kryptografische Operation in Ihrem externen Schlüsselmanager unter Verwendung Ihres externen Schlüssels durchgeführt.

> **Warning**
>
> Der externe Schlüssel ist für den Betrieb des KMS-Schlüssels unerlässlich. Wenn der externe Schlüssel verloren geht oder gelöscht wird, kann der unter dem zugehörigen KMS-Schlüssel verschlüsselte Geheimtext nicht wiederhergestellt werden.

Für externe Schlüsselspeicher muss es sich bei einem externen Schlüssel um einen 256-Bit-AES-Schlüssel handeln, der aktiviert ist und Verschlüsselung und Entschlüsselung durchführen kann. Ausführliche Informationen zu den Anforderungen an externe Schlüssel finden Sie unter Anforderungen an einen KMS-Schlüssel in einem externen Schlüsselspeicher.

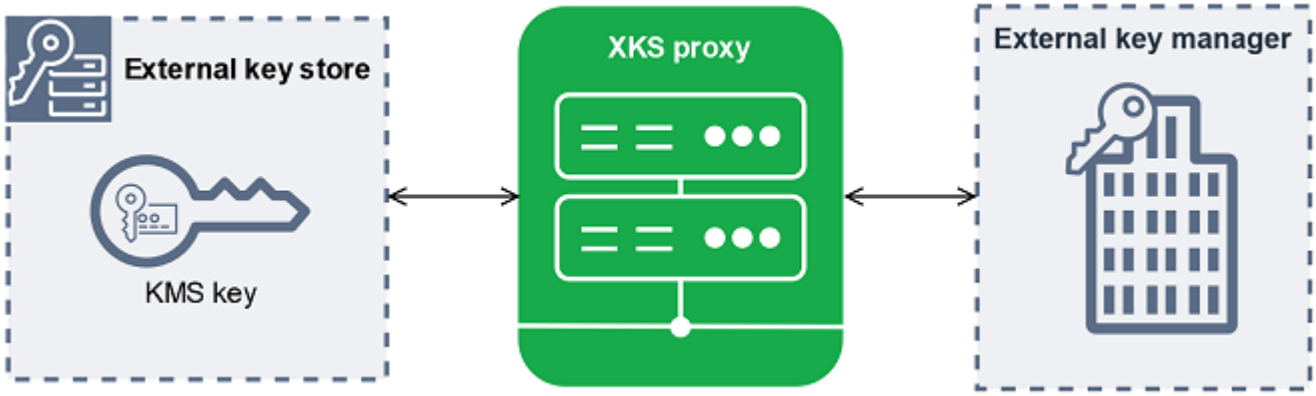
AWS KMS kann keine externen Schlüssel erstellen, löschen oder verwalten. Ihr kryptografisches Schlüsselmaterial verlässt niemals Ihren externen Schlüsselmanager. Wenn Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher erstellen, geben Sie die ID eines externen Schlüssels (`XksKeyId`) an. Sie können die externe Schlüssel-ID, die einem KMS-Schlüssel zugeordnet ist, nicht ändern, obwohl Ihr externer Schlüsselmanager das der externen Schlüssel-ID zugeordnete Schlüsselmaterial rotieren kann.

Zusätzlich zu Ihrem externen Schlüssel enthält ein KMS-Schlüssel in einem externen Schlüsselspeicher auch AWS KMS -Schlüsselmaterial. Durch den KMS-Schlüssel geschützte Daten werden zuerst AWS KMS mithilfe des AWS KMS Schlüsselmaterials und dann von Ihrem externen Schlüsselmanager mithilfe Ihres externen Schlüssels verschlüsselt. Dieser Prozess der doppelten Verschlüsselung stellt sicher, dass der durch Ihren KMS-Schlüssel geschützte Geheimtext immer mindestens so stark ist wie der nur durch AWS KMS geschützte Geheimtext.

Viele kryptografische Schlüssel haben unterschiedliche Arten von Identifikatoren. Wenn Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher erstellen, geben Sie die ID des externen Schlüssels an, den der externe Schlüsselspeicher-Proxy verwendet, um auf den externen Schlüssel zu verweisen. Wenn Sie die falsche Kennung verwenden, schlägt Ihr Versuch, einen KMS-Schlüssel in Ihrem externen Schlüsselspeicher zu erstellen, fehl.

Externer Schlüsselspeicher-Proxy

Der externe Schlüsselspeicher-Proxy („XKS-Proxy“) ist eine kundeneigene und vom Kunden verwaltete Softwareanwendung, die die gesamte Kommunikation zwischen AWS KMS und Ihrem externen Schlüsselmanager vermittelt. Außerdem übersetzt er generische AWS KMS Anfragen in ein Format, das Ihr herstellerspezifischer externer Schlüsselmanager versteht. Für einen externen Schlüsselspeicher ist ein externer Schlüsselspeicher-Proxy erforderlich. Jeder externe Schlüsselspeicher ist mit einem externen Schlüsselspeicher-Proxy verbunden.



AWS KMS kann keine externen Schlüssel erstellen, löschen oder verwalten. Ihr kryptographisches Schlüsselmaterial verlässt zu keiner Zeit Ihren externen Schlüsselmanager. Die gesamte Kommunikation zwischen AWS KMS und Ihrem externen Schlüsselmanager wird über Ihren externen Schlüsselspeicher-Proxy vermittelt. AWS KMS sendet Anfragen an den externen Schlüsselspeicher-Proxy und empfängt Antworten vom externen Schlüsselspeicher-Proxy. Der externe Schlüsselspeicher-Proxy ist dafür verantwortlich AWS KMS , Anfragen von Ihrem externen Schlüsselmanager und Antworten von Ihrem externen Schlüsselmanager zurück an AWS KMS

Sie besitzen und verwalten den externen Schlüsselspeicher-Proxy für Ihren externen Schlüsselspeicher und sind für dessen Wartung und Betrieb verantwortlich. Sie können Ihren externen Schlüsselspeicher-Proxy auf der Grundlage der Open-Source-Proxy-API-Spezifikation für externe Schlüsselspeicher entwickeln, mit der eine Proxyanwendung von einem Anbieter AWS KMS veröffentlicht oder gekauft wird. Ihr externer Schlüsselspeicher-Proxy ist möglicherweise in Ihrem externen Schlüsselmanager enthalten. Zur Unterstützung der Proxyentwicklung werden

AWS KMS außerdem ein Beispiel für einen externen Schlüsselspeicher-Proxy (aws-kms-xks-proxy) und ein Testclient (xks-kms-xksproxy-test-client) bereitgestellt, der überprüft, ob Ihr externer Schlüsselspeicher-Proxy der Spezifikation entspricht.

Zur Authentifizierung verwendet der Proxy AWS KMS serverseitige TLS-Zertifikate. Um sich bei Ihrem Proxy zu authentifizieren, AWS KMS signieren Sie alle Anfragen an Ihren externen Schlüsselspeicher-Proxy mit einem SigV4-Proxy-Authentifizierungsnachweis. Optional kann Ihr Proxy Mutual TLS (mTLS) aktivieren, um zusätzlich sicherzustellen, dass er nur Anfragen von akzeptiert. AWS KMS

Ihr externer Schlüsselspeicher-Proxy muss HTTP/1.1 oder höher und TLS 1.2 oder höher mit mindestens einer der folgenden Verschlüsselungssammlungen unterstützen:

- TLS_AES_256_GCM_SHA384 (TLS 1.3)
- TLS_CHACHA20_POLY1305_SHA256 (TLS 1.3)

> **Note**
>
> Der AWS GovCloud (US) Region unterstützt TLS_CHACHA20_POLY1305_SHA256 nicht.

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)

Verbinden Sie den externen Schlüsselspeicher zunächst mit dem externen Schlüsselspeicher-Proxy, um die KMS-Schlüssel in Ihrem externen Schlüsselspeicher zu erstellen und zu verwenden. Sie können Ihren externen Schlüsselspeicher bei Bedarf auch von seinem Proxy trennen. Wenn Sie dies tun, sind alle KMS-Schlüssel im externen Schlüsselspeicher nicht mehr verfügbar. Sie können nicht für kryptografische Operationen verwendet werden.

Konnektivität des externen Schlüsselspeicher-Proxys

Die Proxykonnektivität für externe Schlüsselspeicher („XKS-Proxykonnektivität“) beschreibt die Methode, die für die Kommunikation mit Ihrem externen Schlüsselspeicher-Proxy verwendet wird. AWS KMS

Sie geben Ihre Proxy-Konnektivitätsoption an, wenn Sie Ihren externen Schlüsselspeicher erstellen, und sie wird zu einer Eigenschaft des externen Schlüsselspeichers. Sie können Ihre Proxy-Konnektivitätsoption ändern, indem Sie die benutzerdefinierte Schlüsselspeichereigenschaft

aktualisieren. Sie müssen jedoch sicherstellen, dass Ihr externer Schlüsselspeicher-Proxy weiterhin auf dieselben externen Schlüssel zugreifen kann.

AWS KMS unterstützt die folgenden Konnektivitätsoptionen:

- Öffentliche Endpunktkonnektivität — AWS KMS sendet Anfragen für Ihren externen Schlüsselspeicher-Proxy über das Internet an einen öffentlichen Endpunkt, den Sie kontrollieren. Diese Option ist einfach zu erstellen und zu verwalten, erfüllt jedoch möglicherweise nicht die Sicherheitsanforderungen für jede Installation.
- VPC-Endpunktservice-Konnektivität — AWS KMS sendet Anfragen an einen Amazon Virtual Private Cloud (Amazon VPC) -Endpunktservice, den Sie erstellen und verwalten. Sie können Ihren externen Schlüsselspeicher-Proxy in Ihrer Amazon VPC hosten oder Ihren externen Schlüsselspeicher-Proxy außerhalb hosten AWS und die Amazon VPC nur für die Kommunikation verwenden.

Einzelheiten zu den Verbindungsoptionen für den externen Schlüsselspeicher-Proxy finden Sie unter Auswählen einer Proxy-Konnektivitätsoption.

Anmeldeinformation für die Proxy-Authentifizierung des externen Schlüsselspeichers

Um sich bei Ihrem externen Schlüsselspeicher-Proxy zu authentifizieren, AWS KMS signieren Sie alle Anfragen an Ihren externen Schlüsselspeicher-Proxy mit einem Signature V4 (SigV4) -Authentifizierungsnachweis. Sie erstellen und verwalten die Authentifizierungsdaten auf Ihrem Proxy und geben diese Anmeldeinformationen dann an, AWS KMS wenn Sie Ihren externen Speicher erstellen.

> **Note**
>
> Die SigV4-Anmeldeinformationen, die zum Signieren von Anfragen an den XKS-Proxy AWS KMS verwendet werden, haben nichts mit irgendwelchen SigV4-Anmeldeinformationen zu den Prinzipalen in Ihrem zu tun. AWS Identity and Access Management AWS-Konten Verwenden Sie keine IAM-SigV4-Anmeldeinformationen für Ihren externen Schlüsselspeicher-Proxy wieder.

Alle Anmeldeinformationen für die Proxy-Authentifizierung bestehen aus zwei Teilen. Sie müssen beide Teile angeben, wenn Sie einen externen Schlüsselspeicher erstellen oder die Anmeldeinformationen für die Authentifizierung für Ihren externen Schlüsselspeicher aktualisieren.

- Zugriffsschlüssel-ID: Identifiziert den geheimen Zugriffsschlüssel. Sie können diese ID in Klartext angeben.
- Geheimer Zugriffsschlüssel: Der geheime Teil der Anmeldeinformationen. AWS KMS verschlüsselt den geheimen Zugriffsschlüssel in den Anmeldeinformationen, bevor er gespeichert wird.

Sie können die Einstellung für die Anmeldeinformation jederzeit bearbeiten, z. B. wenn Sie falsche Werte eingeben, wenn Sie Ihre Anmeldeinformation auf dem Proxy ändern oder wenn Ihr Proxy die Anmeldeinformation rotiert. Technische Informationen zur AWS KMS Authentifizierung beim externen Schlüsselspeicher-Proxy finden Sie unter Authentifizierung in der API-Spezifikation für den AWS KMS externen Schlüsselspeicher-Proxy.

Damit Sie Ihre Anmeldeinformationen rotieren können, ohne die AWS-Services Verwendung von KMS-Schlüsseln in Ihrem externen Schlüsselspeicher zu unterbrechen, empfehlen wir, dass der externe Schlüsselspeicher-Proxy mindestens zwei gültige Authentifizierungsinformationen für unterstützt. AWS KMS Auf diese Weise wird sichergestellt, dass Ihre vorherige Anmeldeinformation weiterhin funktioniert, während Sie Ihre neue Anmeldeinformation für AWS KMS bereitstellen.

Damit Sie das Alter Ihrer Proxyauthentifizierungsdaten verfolgen können, AWS KMS definiert eine CloudWatch Amazon-Metrik, XksProxyCredentialAge. Sie können diese Metrik verwenden, um einen CloudWatch Alarm auszulösen, der Sie benachrichtigt, wenn das Alter Ihrer Anmeldeinformationen einen von Ihnen festgelegten Schwellenwert erreicht.

Um zusätzlich sicherzustellen, dass Ihr externer Schlüsselspeicher-Proxy nur auf AWS KMS antwortet, unterstützen einige externe Schlüssel-Proxys mTLS (mutual Transport Layer Security). Details hierzu finden Sie unter mTLS-Authentifizierung (optional).

Proxy-APIs

Um einen AWS KMS externen Schlüsselspeicher zu unterstützen, muss ein externer Schlüsselspeicher-Proxy die erforderlichen Proxy-APIs implementieren, wie in der Spezifikation für AWS KMS externe Schlüsselspeicher-Proxy-APIs beschrieben. Diese Proxy-API-Anfragen sind die einzigen Anfragen, die AWS KMS an den Proxy gesendet werden. Auch wenn Sie diese Anforderungen nie direkt senden, kann das Wissen darüber Ihnen helfen, Probleme zu beheben, die mit Ihrem externen Schlüsselspeicher oder seinem Proxy auftreten könnten. Nimmt beispielsweise AWS KMS Informationen über die Latenz und die Erfolgsraten dieser API-Aufrufe in seine CloudWatch Amazon-Metriken für externe Schlüsselgeschäfte auf. Details hierzu finden Sie unter Überwachung eines externen Schlüsselspeichers.

In der folgenden Tabelle sind die einzelnen Proxy-APIs aufgeführt und beschrieben. Dazu gehören auch die AWS KMS Vorgänge, die einen Aufruf der Proxy-API auslösen, sowie alle mit der Proxy-API verbundenen AWS KMS Betriebsausnahmen.

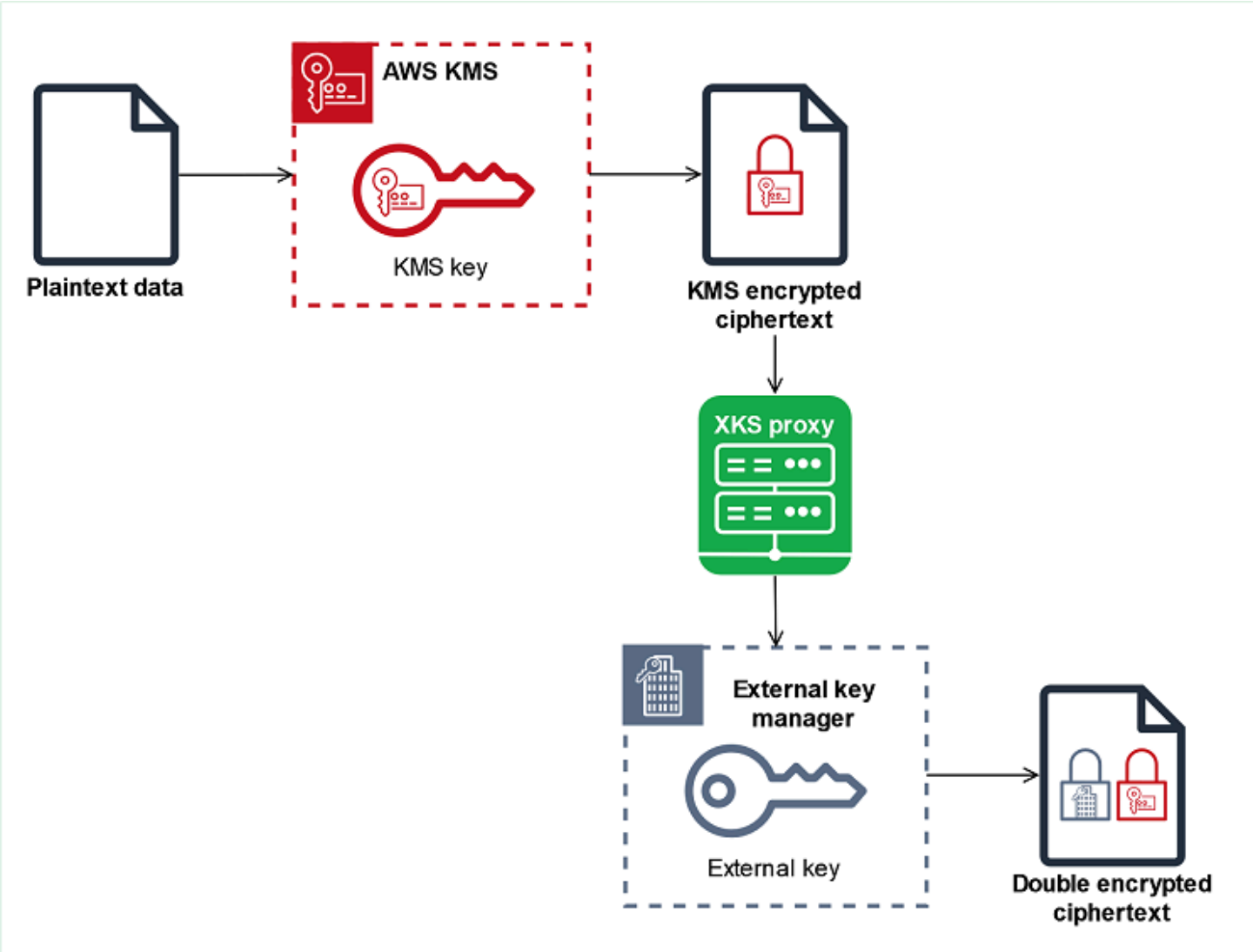
Proxy-API	Beschreibung	Verwandte AWS KMS Operationen
Decrypt	AWS KMS sendet den zu entschlüsselnden Chiffretext und die ID des zu verwendenden externen Schlüssels. Der erforderliche Verschlüsselungsalgorithmus ist AES_GCM.	Entschlüsseln, ReEncrypt
Encrypt	AWS KMS sendet zu verschlüsselnde Daten und die ID des zu verwenden den externen Schlüssels. Der erforderliche Verschlüsselungsalgorithmus ist AES_GCM.	Verschlüsseln,, GenerateDataKey, GenerateDataKeyWithoutPlaintextReEncrypt
GetHealthStatus	AWS KMS fordert Informationen über den Status des Proxys und Ihres externen Schlüsselmanagers an. Der Status eines jeden externen Schlüsselmanagers kann einer der folgenden sein. • `Active`: Fehlerfrei; kann den Datenverkehr abwickeln • `Degraded`: Fehlerhaft, kann aber den Datenverkehr abwickeln • `Unavailable` : Fehlerhaft; kann keinen Datenverkehr abwickeln	CreateCustomKeyStore(für öffentliche Endpunktkonnektivität), ConnectCustomKeyStore(für VPC-Endpunktdienstkonnektivität) Wenn alle externen Schlüsselmanager-Instances `Unavailable` sind, schlagen Versuche, den Schlüsselspeicher zu erstellen oder eine Verbindung herzustellen, mit `XksProxyUriUnreachableException` fehl.
GetKeyMetadata	AWS KMS fordert Informationen über den externen Schlüssel an, der einem KMS-Schlüssel in Ihrem externen Schlüsselspeicher zugeordnet ist.	CreateKey Wenn die Schlüsselspezifikation nicht `AES_256` ist oder die Schlüsselverwendung nicht `[ENCRYPT,`

Proxy-API	Beschreibung	Verwandte AWS KMS Operationen
	Die Antwort enthält die Schlüsselspezifikation (`AES_256`), die Schlüsselverwendung (`[ENCRYPT, DECRYPT]`) und die Angabe, ob der externe Schlüssel `ENABLED` oder `DISABLED` ist.	`DECRYPT]` ist oder der Status `DISABLED` ist, schlägt der Vorgang `CreateKey` mit `XksKeyInvalidConfigurationException` fehl.

Doppelte Verschlüsselung

Daten, die mit einem KMS-Schlüssel in einem externen Schlüsselspeicher verschlüsselt wurden, werden zweimal verschlüsselt. AWS KMS Verschlüsselt zunächst die Daten mit AWS KMS Schlüsselmaterial, das für den KMS-Schlüssel spezifisch ist. Dann wird der mit AWS KMS verschlüsselte Geheimtext von Ihrem externen Schlüsselmanager mit Ihrem externen Schlüssel verschlüsselt. Dieser Vorgang wird als doppelte Verschlüsselung bezeichnet.

Die doppelte Verschlüsselung stellt sicher, dass Daten, die mit einem KMS-Schlüssel in einem externen Schlüsselspeicher verschlüsselt werden, mindestens so sicher sind wie mit einem Standard-KMS-Schlüssel verschlüsselter Geheimtext. Es schützt auch Ihren Klartext bei der Übertragung von AWS KMS zu Ihrem externen Schlüsselspeicher-Proxy. Bei der doppelten Verschlüsselung behalten Sie die volle Kontrolle über Ihre Geheimtexte. Wenn Sie AWS den Zugriff auf Ihren externen Schlüssel über Ihren externen Proxy dauerhaft entziehen, wird jeglicher in AWS verbleibende Geheimtext effektiv kryptografisch zerlegt.



Jeder KMS-Schlüssel verfügt in einem externen Schlüsselspeicher über zwei kryptografische Unterstützungsschlüssel, um eine doppelte Verschlüsselung zu ermöglichen:

- Ein AWS KMS Schlüsselmaterial, das nur für den KMS-Schlüssel gilt. Dieses Schlüsselmaterial wird generiert und nur in AWS KMS FIPS 140-2 Security Level 3-zertifizierten Hardware-Sicherheitsmodulen (HSMs) verwendet.
- Ein externer Schlüssel in Ihrem externen Schlüsselmanager.

Die doppelte Verschlüsselung hat die folgenden Auswirkungen:

- AWS KMS kann keinen Chiffretext entschlüsseln, der mit einem KMS-Schlüssel in einem externen Schlüsselspeicher verschlüsselt wurde, ohne über Ihren externen Schlüsselspeicher-Proxy auf Ihre externen Schlüssel zuzugreifen.

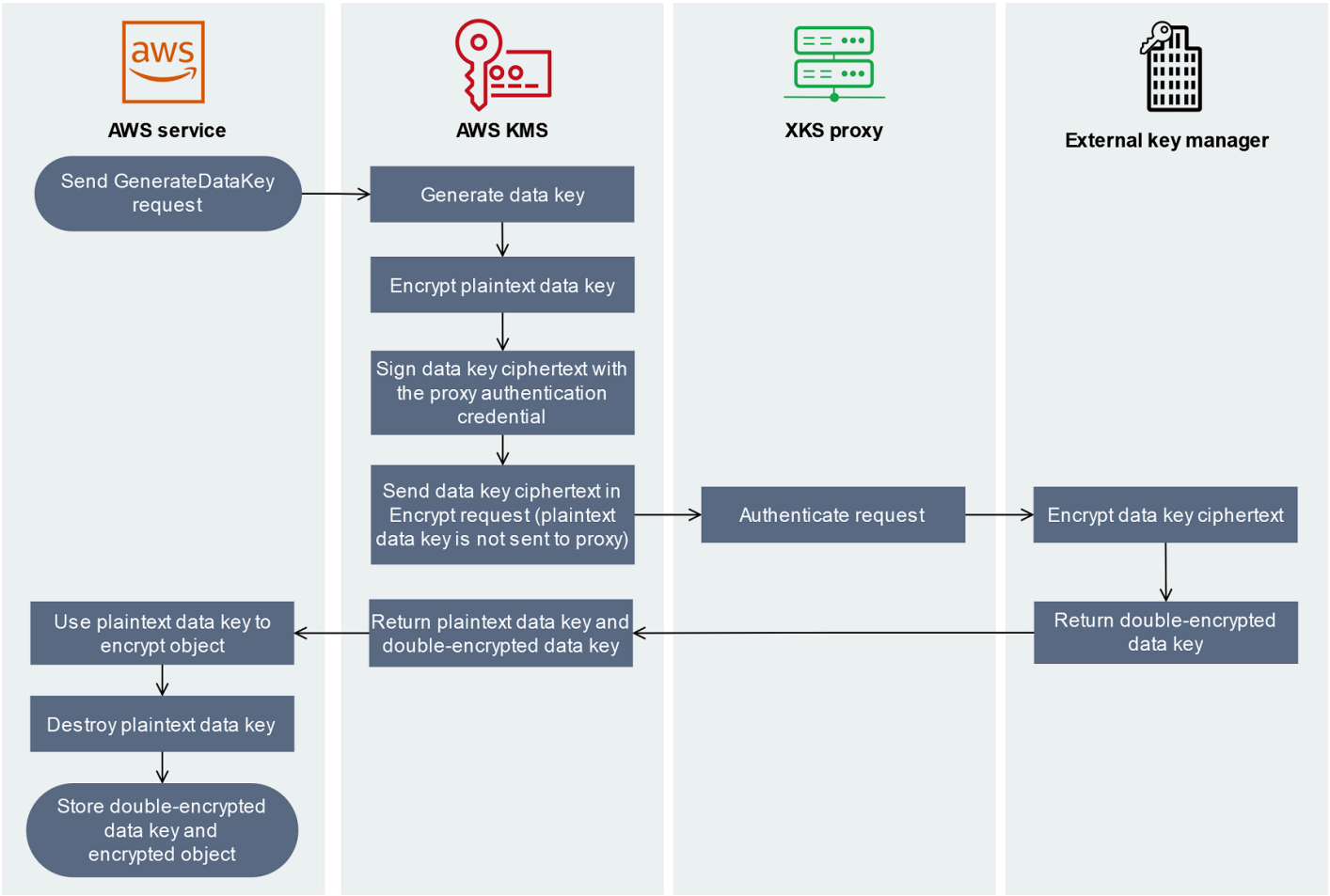
- Sie können keinen mit einem KMS-Schlüssel verschlüsselten Chiffretext in einem externen Schlüsselspeicher außerhalb von entschlüsseln AWS, selbst wenn Sie über dessen externes Schlüsselmaterial verfügen.
- Sie können einen KMS-Schlüssel, der aus einem externen Schlüsselspeicher gelöscht wurde, nicht wiederherstellen, selbst wenn Sie über sein externes Schlüsselmaterial verfügen. Jeder KMS-Schlüssel verfügt über eindeutige Metadaten, die er in den symmetrischen Geheimtext einfügt. Ein neuer KMS-Schlüssel wäre nicht in der Lage, einen mit dem ursprünglichen Schlüssel verschlüsselten Geheimtext zu entschlüsseln, selbst wenn er dasselbe externe Schlüsselmaterial verwenden würde.

Ein Beispiel für doppelte Verschlüsselung in der Praxis finden Sie unter Funktionsweise externer Schlüsselspeicher.

Funktionsweise externer Schlüsselspeicher

Ihr externer Schlüsselspeicher, der externe Schlüsselspeicher-Proxy und der externe Schlüsselmanager arbeiten zusammen, um Ihre AWS -Ressourcen zu schützen. Das folgende Verfahren zeigt den Verschlüsselungsworkflow eines typischen AWS-Service , bei dem jedes Objekt mit einem eindeutigen Datenschlüssel verschlüsselt wird, der durch einen KMS-Schlüssel geschützt ist. In diesem Fall haben Sie zum Schutz des Objekts einen KMS-Schlüssel in einem externen Schlüsselspeicher gewählt. Das Beispiel zeigt, wie doppelte Verschlüsselung AWS KMS verwendet wird, um den Datenschlüssel während der Übertragung zu schützen und sicherzustellen, dass Chiffretext, der durch einen KMS-Schlüssel in einem externen Schlüsselspeicher generiert wird, immer mindestens so stark ist wie Chiffretext, der mit einem standardmäßigen symmetrischen KMS-Schlüssel verschlüsselt wurde, in dem Schlüsselmaterial enthalten ist. AWS KMS

Die Verschlüsselungsmethoden, die von den einzelnen Integrationen verwendet werden, sind unterschiedlich AWS-Service . AWS KMS Weitere Informationen finden Sie unter dem Thema zum Datenschutz im Kapitel über Sicherheit in der AWS-Service -Dokumentation.



1. Sie fügen Ihrer AWS-Service Ressource ein neues Objekt hinzu. Um das Objekt zu verschlüsseln, AWS-Service sendet der eine GenerateDataKeyAnfrage zur AWS KMS Verwendung eines KMS-Schlüssels in Ihrem externen Schlüsselspeicher.
2. AWS KMS generiert einen symmetrischen 256-Bit-Datenschlüssel und bereitet den Versand einer Kopie des Klartext-Datenschlüssels über Ihren externen Schlüsselspeicher-Proxy an Ihren externen Schlüsselmanager vor. AWS KMS beginnt den doppelten Verschlüsselungsprozess, indem der Klartext-Datenschlüssel mit dem Schlüsselmaterial verschlüsselt wird, das dem AWS KMS KMS-Schlüssel im externen Schlüsselspeicher zugeordnet ist.
3. AWS KMS sendet eine Verschlüsselungsanforderung an den externen Schlüsselspeicher-Proxy, der dem externen Schlüsselspeicher zugeordnet ist. Die Anforderung enthält den zu verschlüsselnden Chiffretext des Datenschlüssels und die ID des externen Schlüssels, der dem KMS-Schlüssel zugeordnet ist. AWS KMS signiert die Anfrage mit den Proxyauthentifizierungsdaten für Ihren externen Schlüsselspeicher-Proxy.

Die Klartextkopie des Datenschlüssels wird nicht an den externen Schlüsselspeicher-Proxy gesendet.

4. Der externe Schlüsselspeicher-Proxy authentifiziert die Anforderung und leitet die Verschlüsselungsanforderung dann an Ihren externen Schlüsselmanager weiter.

   Einige externe Schlüsselspeicher-Proxys implementieren auch eine optionale Autorisierungsrichtlinie, die es nur ausgewählten Prinzipalen erlaubt, unter bestimmten Bedingungen Operationen durchzuführen.

5. Ihr externer Schlüsselmanager verschlüsselt den Geheimtext des Datenschlüssels mit dem angegebenen externen Schlüssel. Der externe Schlüsselmanager gibt den doppelt verschlüsselten Datenschlüssel an Ihren externen Schlüsselspeicher-Proxy zurück, der ihn an AWS KMS zurückgibt.
6. AWS KMS gibt den Klartext-Datenschlüssel und die doppelt verschlüsselte Kopie dieses Datenschlüssels an den zurück. AWS-Service
7. Der AWS-Service verwendet den Klartext-Datenschlüssel, um das Ressourcenobjekt zu verschlüsseln, zerstört den Klartext-Datenschlüssel und speichert den verschlüsselten Datenschlüssel zusammen mit dem verschlüsselten Objekt.

   In manchen AWS-Services Fällen wird der Klartext-Datenschlüssel möglicherweise zwischengespeichert, um ihn für mehrere Objekte zu verwenden oder um ihn wiederzuverwenden, während die Ressource verwendet wird. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Um das verschlüsselte Objekt zu entschlüsseln, AWS-Service müssen sie den verschlüsselten Datenschlüssel AWS KMS in einer Entschlüsselungsanforderung an sie zurücksenden. Um den verschlüsselten Datenschlüssel zu entschlüsseln, AWS KMS müssen Sie den verschlüsselten Datenschlüssel mit der ID des externen Schlüssels an Ihren externen Schlüsselspeicher-Proxy zurücksenden. Wenn die Entschlüsselungsanforderung an den externen Schlüsselspeicher-Proxy aus irgendeinem Grund fehlschlägt, AWS KMS kann der verschlüsselte Datenschlüssel nicht entschlüsselt werden, und das verschlüsselte Objekt AWS-Service kann nicht entschlüsselt werden.

Steuern des Zugriffs auf Ihren externen Schlüsselspeicher

Alle AWS KMS-Zugriffskontrollfunktionen – Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen –, die Sie mit Standard-KMS-Schlüsseln verwenden, funktionieren auf die gleiche Weise für KMS-Schlüssel in einem externen Schlüsselspeicher. Sie können IAM-Richtlinien verwenden, um den

Zugriff auf die API-Operationen zur Erstellung und Verwaltung externer Schlüsselspeicher zu steuern. Sie verwenden IAM-Richtlinien und Schlüsselrichtlinien, um den Zugriff auf die AWS KMS keys in Ihrem externen Schlüsselspeicher zu steuern. Sie können auch Service-Kontrollrichtlinien für Ihre AWS-Organisation und VPC-Endpunktrichtlinien verwenden, um den Zugriff auf KMS-Schlüssel in Ihrem externen Schlüsselspeicher zu kontrollieren.

Sie sollten Benutzern und Rollen ausschließlich die Berechtigungen gewähren, die sie für die Aufgaben benötigen, die sie voraussichtlich ausführen werden.

Themen

Autorisierung externer Schlüsselspeichermanager

Prinzipale, die einen externen Schlüsselspeicher erstellen und verwalten, benötigen Berechtigungen für die benutzerdefinierten Schlüsselspeicheroperationen. Die folgende Liste beschreibt die Mindestberechtigungen, die Manager externer Schlüsselspeicher benötigen. Da es sich bei einem benutzerdefinierten Schlüsselspeicher nicht um eine AWS-Ressource handelt, können Sie einem externen Schlüsselspeicher keine Berechtigungen für Prinzipale in anderen AWS-Konten geben.

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms:DeleteCustomKeyStore`

Prinzipale, die einen externen Schlüsselspeicher erstellen, benötigen die Berechtigung, die externen Schlüsselspeicherkomponenten zu erstellen und zu konfigurieren. Prinzipale können externe Schlüsselspeicher nur in ihren eigenen Konten erstellen. Um einen externen Schlüsselspeicher

mit der Konnektivität eines VPC-Endpunkt-Services zu erstellen, müssen die Prinzipale über die Berechtigung verfügen, die folgenden Komponenten zu erstellen:

- Eine Amazon VPC
- Öffentliche und private Subnetze
- Einen Network Load Balancer und eine Zielgruppe
- Einen VPC-Endpunkt-Service von Amazon

Einzelheiten finden Sie unter Identity and Access Management für Amazon VPC, Identity and Access Management für VPC-Endpunkte und VPC-Endpunktservices und Elastic Load Balancing-API-Berechtigungen.

Autorisierung von Benutzern von KMS-Schlüsseln in externen Schlüsselspeichern

Prinzipale, die AWS KMS keys in Ihrem externen Schlüsselspeicher erstellen und verwalten, benötigen die gleichen Berechtigungen wie Prinzipale, die KMS-Schlüssel in AWS KMS erstellen und verwalten. Die Standard-Schlüsselrichtlinie für KMS-Schlüssel in einem externen Schlüsselspeicher ist identisch mit der Standard-Schlüsselrichtlinie für KMS-Schlüssel in AWS KMS. Attributbasierte Zugriffskontrolle (ABAC), das Tags und Aliasse verwendet, um den Zugriff auf KMS-Schlüssel zu steuern, ist auch für KMS-Schlüssel in externen Schlüsselspeichern wirksam.

Prinzipale, die die KMS-Schlüssel in Ihrem benutzerdefinierten Schlüsselspeicher für kryptografische Operationen verwenden, benötigen die Berechtigung zum Ausführen der kryptografischen Operation mit dem KMS-Schlüssel, z. B. kms:Decrypt. Sie können diese Berechtigungen in einer IAM-Richtlinie oder Schlüsselrichtlinie bereitstellen. Sie benötigen keine zusätzlichen Berechtigungen, um einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden zu können.

Um eine Berechtigung festzulegen, die nur für KMS-Schlüssel in einem externen Schlüsselspeicher gilt, verwenden Sie die Richtlinienbedingung `kms:KeyOrigin` mit einem Wert von `EXTERNAL_KEY_STORE`. Sie können diese Bedingung verwenden, um die kms:CreateKey-Berechtigung oder jede Berechtigung einzuschränken, die für eine KMS-Schlüsselressource spezifisch ist. Die folgende IAM-Richtlinie erlaubt es beispielsweise der Identität, der sie zugeordnet ist, die angegebenen Operationen für alle KMS-Schlüssel des Kontos aufzurufen, vorausgesetzt, die KMS-Schlüssel befinden sich in einem externen Schlüsselspeicher. Beachten Sie, dass Sie die Berechtigung auf KMS-Schlüssel in einem externen Schlüsselspeicher und KMS-Schlüssel in einem AWS-Konto beschränken können, aber nicht auf einen bestimmten externen Schlüsselspeicher im Konto.

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

Autorisierung von AWS KMS zur Kommunikation mit Ihrem externen Schlüsselspeicher-Proxy

AWS KMS kommuniziert mit Ihrem externen Schlüsselmanager nur über den von Ihnen bereitgestellten externen Schlüsselspeicher-Proxy. AWS KMS authentifiziert sich bei Ihrem Proxy, indem es seine Anforderungen unter Verwendung des Prozesses Signature Version 4 (SigV4) mit der von Ihnen angegebenen Anmeldeinformation für die Proxy-Authentifizierung des externen Schlüsselspeichers signiert. Wenn Sie eine Konnektivität eines öffentlichen Endpunkts für Ihren externen Schlüsselspeicher-Proxy verwenden, benötigt AWS KMS keine zusätzlichen Berechtigungen.

Wenn Sie jedoch die Konnektivität eines VPC-Endpunkt-Service verwenden, müssen Sie AWS KMS die Erlaubnis erteilen, einen Schnittstellenendpunkt zu Ihrem VPC-Endpunkt-Service von Amazon zu erstellen. Diese Berechtigung ist unabhängig davon erforderlich, ob sich der externe Schlüsselspeicher-Proxy in Ihrer VPC befindet oder ob der externe Schlüsselspeicher-Proxy sich an einem anderen Ort befindet, aber den VPC-Endpunkt-Service zur Kommunikation mit AWS KMS verwendet.

Um AWS KMS zu erlauben, einen Schnittstellenendpunkt zu erstellen, verwenden Sie die Amazon-VPC-Konsole oder die -ModifyVpcEndpointServicePermissionsOperation. Erlauben Sie Berechtigungen für den folgenden Prinzipal: `cks.kms.<region>.amazonaws.com`.

Mit dem folgenden AWS CLI-Befehl kann AWS KMS beispielsweise eine Verbindung mit dem angegebenen VPC-Endpunkt-Service in der Region USA West (Oregon) (us-west-2) herstellen.

Ersetzen Sie vor der Verwendung dieses Befehls die Amazon-VPC-Service-ID und AWS-Region durch gültige Werte für Ihre Konfiguration.

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Um diese Berechtigung zu entfernen, verwenden Sie die Amazon-VPC-Konsole oder die ModifyVpcEndpointServicePermissions mit dem `RemoveAllowedPrincipals` Parameter .

Proxy-Autorisierung für externen Schlüsselspeicher (optional)

Einige externe Schlüsselspeicher-Proxys implementieren Autorisierungsanforderungen für die Verwendung ihrer externen Schlüssel. Ein externer Schlüsselspeicher-Proxy ist berechtigt, aber nicht verpflichtet, ein Autorisierungsschema zu entwerfen und zu implementieren, das es bestimmten Benutzern erlaubt, bestimmte Operationen nur unter bestimmten Bedingungen anzufordern. Beispielsweise könnte ein Proxy so konfiguriert sein, dass er Benutzer:in A die Verschlüsselung mit einem bestimmten externen Schlüssel erlaubt, aber nicht die Entschlüsselung mit diesem Schlüssel.

Die Proxy-Autorisierung ist unabhängig von der SigV4-basierten Proxy-Authentifizierung, die AWS KMS für alle externen Schlüsselspeicher-Proxys benötigt. Sie ist auch unabhängig von den Schlüsselrichtlinien, IAM-Richtlinien und Berechtigungen, die den Zugriff auf Operationen zulassen, die den externen Schlüsselspeicher oder seine KMS-Schlüssel betreffen.

AWS KMS nimmt Metadaten in jede Proxy-API-Anforderung auf, einschließlich des Aufrufers, des KMS-Schlüssels, der AWS KMS-Operation und des AWS-Service (falls vorhanden), um die Autorisierung durch den externen Schlüsselspeicher-Proxy zu ermöglichen. Die Anforderungsmetadaten für Version 1 (v1) der externen Schlüssel-Proxy-API lauten wie folgt.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Sie können Ihren Proxy zum Beispiel so konfigurieren, dass er Anforderungen von einem bestimmten Prinzipal (`awsPrincipalArn`) zulässt, aber nur, wenn die Anforderung im Namen des Prinzipals von einem bestimmten AWS-Service (`kmsViaService`) gestellt wird.

Wenn die Proxy-Autorisierung fehlschlägt, scheitert der entsprechende AWS KMS-Vorgang mit einer Meldung, die den Fehler erklärt. Details hierzu finden Sie unter Probleme mit der Proxy-Autorisierung.

mTLS-Authentifizierung (optional)

Damit Ihr externer Schlüsselspeicher-Proxy Anforderungen von AWS KMS authentifizieren kann, signiert AWS KMS alle Anforderungen an Ihren externen Schlüsselspeicher-Proxy mit der Proxy-Authentifizierungsanmeldeinformation Signature V4 (SigV4) für Ihren externen Schlüsselspeicher.

Um zusätzlich sicherzustellen, dass Ihr externer Schlüsselspeicher-Proxy nur auf AWS KMS-Anforderungen antwortet, unterstützen einige externe Schlüsselspeicher-Proxys mTLS (mutual Transport Layer Security), bei der beide Parteien einer Transaktion Zertifikate zur gegenseitigen Authentifizierung verwenden. mTLS fügt der serverseitigen Authentifizierung, die Standard-TLS bietet, eine clientseitige Authentifizierung hinzu, bei der der externe Schlüsselspeicher-Proxy-Server den AWS KMS-Client authentifiziert. In dem seltenen Fall, dass Ihre Proxy-Authentifizierungsanmeldeinformation kompromittiert wird, verhindert mTLS, dass ein Dritter erfolgreiche API-Anforderungen an den externen Schlüsselspeicher-Proxy stellt.

Um mTLS zu implementieren, konfigurieren Sie Ihren externen Schlüsselspeicher-Proxy so, dass er nur clientseitige TLS-Zertifikate mit den folgenden Eigenschaften akzeptiert:

- Der Subject Common Name auf dem TLS-Zertifikat muss `cks.kms.<Region>.amazonaws.com` lauten, zum Beispiel `cks.kms.eu-west-3.amazonaws.com`.
- Das Zertifikat muss mit einer Zertifizierungsstelle verknüpft sein, die mit Amazon Trust Services verbunden ist.

Planen eines externen Schlüsselspeichers

Bevor Sie einen externen Schlüsselspeicher erstellen, wählen Sie die Konnektivitätsoption aus, die bestimmt, wie AWS KMS mit den Komponenten Ihres externen Schlüsselspeichers kommuniziert. Der Rest des Planungsprozesses hängt davon ab, welche Konnektivitätsoption Sie auswählen.

Weitere Informationen:

- Sehen Sie sich den Prozess zum Erstellen eines externen Schlüsselspeichers an, einschließlich der Informationen zum Erfüllen der Voraussetzungen. Er stellt sicher, dass Sie über alle benötigten Komponenten verfügen, wenn Sie einen externen Schlüsselspeicher erstellen.
- Erfahren Sie, wie Sie den Zugriff auf Ihren externen Schlüsselspeicher steuern, einschließlich der Berechtigungen, die Administratoren und Benutzer von externen Schlüsselspeicheradministratoren benötigen.
- Erfahren Sie mehr über die Amazon- CloudWatch Metriken und -Dimensionen, die für externe Schlüsselspeicher AWS KMS aufzeichnet. Wir empfehlen Ihnen dringend, Alarme zur Überwachung Ihres externen Schlüsselspeichers zu erstellen, damit Sie erste Anzeichen von Leistungs- und Betriebsproblemen erkennen.

Auswählen einer Proxy-Konnektivitätsoption

Wenn Sie einen externen Schlüsselspeicher erstellen, müssen Sie festlegen, wie AWS KMS mit Ihrem externen Schlüsselspeicher-Proxy kommuniziert. Diese Entscheidung bestimmt, welche Komponenten Sie benötigen und wie Sie diese konfigurieren. AWS KMS unterstützt die folgenden Konnektivitätsoptionen. Wählen Sie die Option aus, die Ihren Leistungs- und Sicherheitszielen entspricht.

Bevor Sie beginnen, vergewissern Sie sich, dass Sie einen externen Schlüsselspeicher benötigen. Die meisten Kunden können KMS-Schlüssel verwenden, die durch AWS KMS-Schlüsselmaterial gesichert sind.

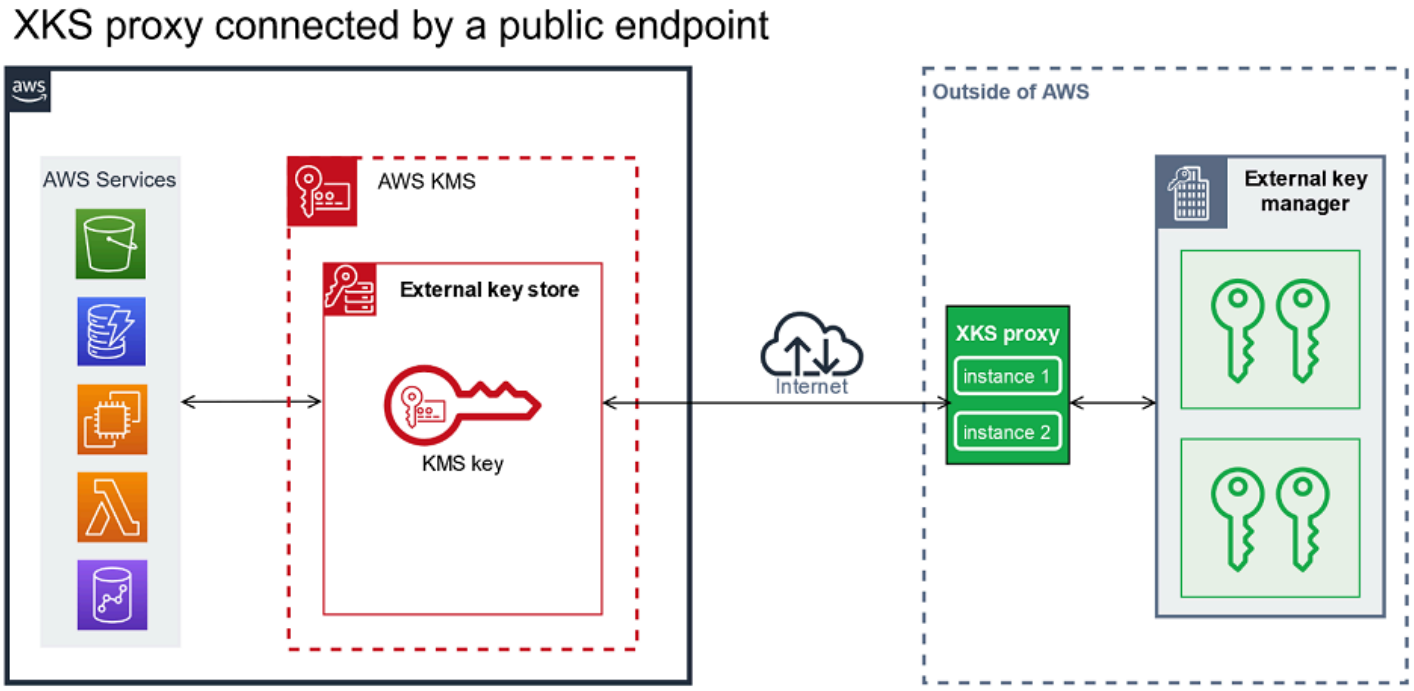
> **Note**
>
> Wenn Ihr externer Schlüsselspeicher-Proxy in Ihren externen Schlüsselmanager integriert ist, ist die Konnektivität möglicherweise vordefiniert. Hilfreiche Informationen finden Sie in der Dokumentation für Ihren externen Schlüsselmanager oder externen Schlüsselspeicher-Proxy.

Sie können die Konnektivitätsoption für Ihren externen Schlüsselspeicher-Proxy auch dann ändern, wenn der entsprechende Schlüsselspeicher aktiv ist. Dieser Vorgang muss jedoch sorgfältig geplant und ausgeführt werden, um Unterbrechungen zu minimieren, Fehler zu vermeiden und einen unterbrechungsfreien Zugriff auf die kryptografischen Schlüssel sicherzustellen, mit denen Ihre Daten verschlüsselt werden.

Konnektivität eines öffentlichen Endpunkts

AWS KMS stellt mithilfe eines öffentlichen Endpunkts über das Internet eine Verbindung mit dem externen Schlüsselspeicher-Proxy (XKS-Proxy) her.

Diese Konnektivitätsoption ist einfacher einzurichten und zu verwalten und eignet sich gut für einige Schlüsselverwaltungsmodelle. Sie entspricht jedoch möglicherweise nicht den Sicherheitsanforderungen einiger Organisationen.



Voraussetzungen

Wenn Sie sich für die Konnektivität eines öffentlichen Endpunkts entscheiden, werden die folgenden Komponenten benötigt.

- Ihr externer Schlüsselspeicher-Proxy muss unter einem öffentlich routingfähigen Endpunkt erreichbar sein.
- Sie können einen öffentlichen Endpunkt für mehrere externe Schlüsselspeicher verwenden, sofern diese unterschiedliche Werte für den Proxy-URI-Pfad nutzen.
- Für einen externen Schlüsselspeicher mit der Konnektivität eines öffentlichen Endpunkts und für einen externen Schlüsselspeicher mit der Konnektivität eines VPC-Endpunktservice in derselben AWS-Region können Sie nicht denselben Endpunkt verwenden, selbst wenn sich die Schlüsselspeicher in verschiedenen AWS-Konten befinden.

- Sie benötigen ein TLS-Zertifikat von einer öffentlichen Zertifizierungsstelle, die für externe Schlüsselspeicher unterstützt wird. Eine Liste finden Sie unter Vertrauenswürdige Zertifizierungsstellen.

  Der Subject Common Name (CN) auf dem TLS-Zertifikat muss mit dem Domainnamen im Proxy-URI-Endpunkt für den externen Schlüsselspeicher-Proxy identisch sein. Ist der öffentliche Endpunkt beispielsweise `https://myproxy.xks.example.com`, muss der CN auf dem TLS-Zertifikat `myproxy.xks.example.com` oder `*.xks.example.com` lauten.
- Falls sich Firewalls zwischen AWS KMS und dem externen Schlüsselspeicher-Proxy befinden, stellen Sie sicher, dass diese den Verkehr zu und von Port 443 auf dem Proxy zulassen. AWS KMS kommuniziert auf Port 443. Dieser Wert kann nicht konfiguriert werden.

Alle Anforderungen für einen externen Schlüsselspeicher finden Sie in den Informationen zum Erfüllen der Voraussetzungen.

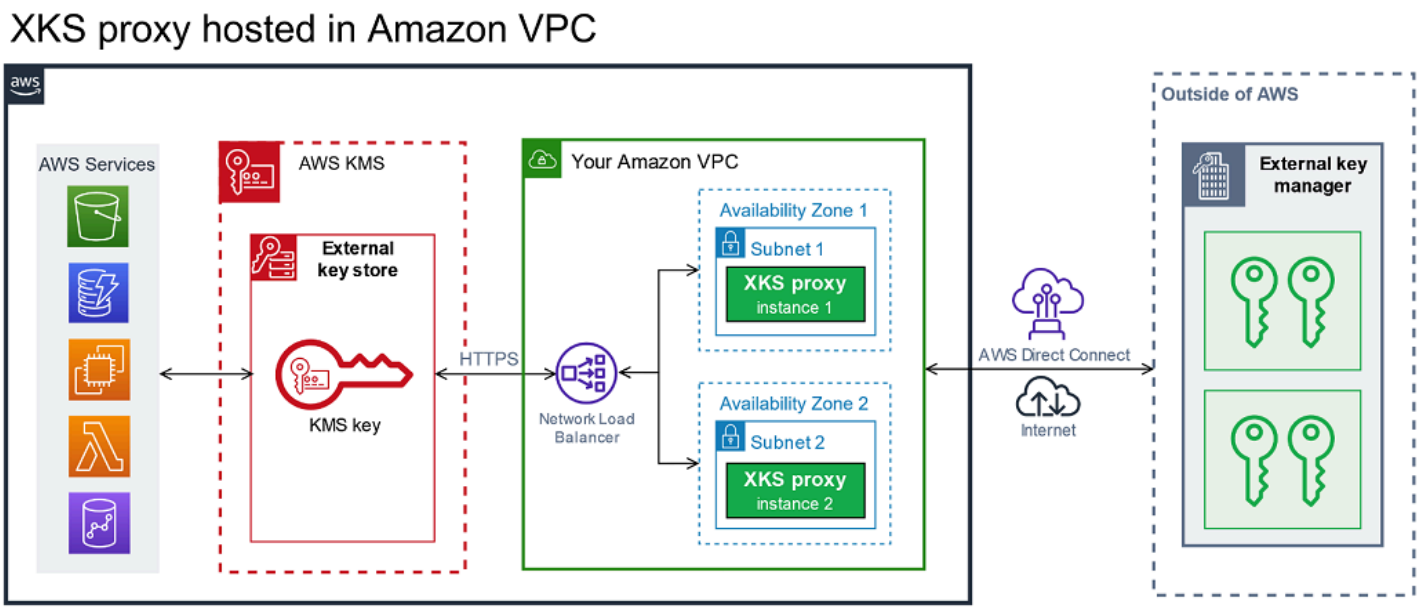
Konnektivität eines VPC-Endpunktservice

AWS KMS stellt eine Verbindung mit dem externen Schlüsselspeicher-Proxy (XKS-Proxy) her, indem ein Schnittstellenendpunkt zu einem Amazon VPC-Endpunktservice erstellt wird, den Sie erstellen und konfigurieren. Sie müssen den VPC-Endpunktservice erstellen und Ihre VPC mit Ihrem externen Schlüsselmanager verbinden.

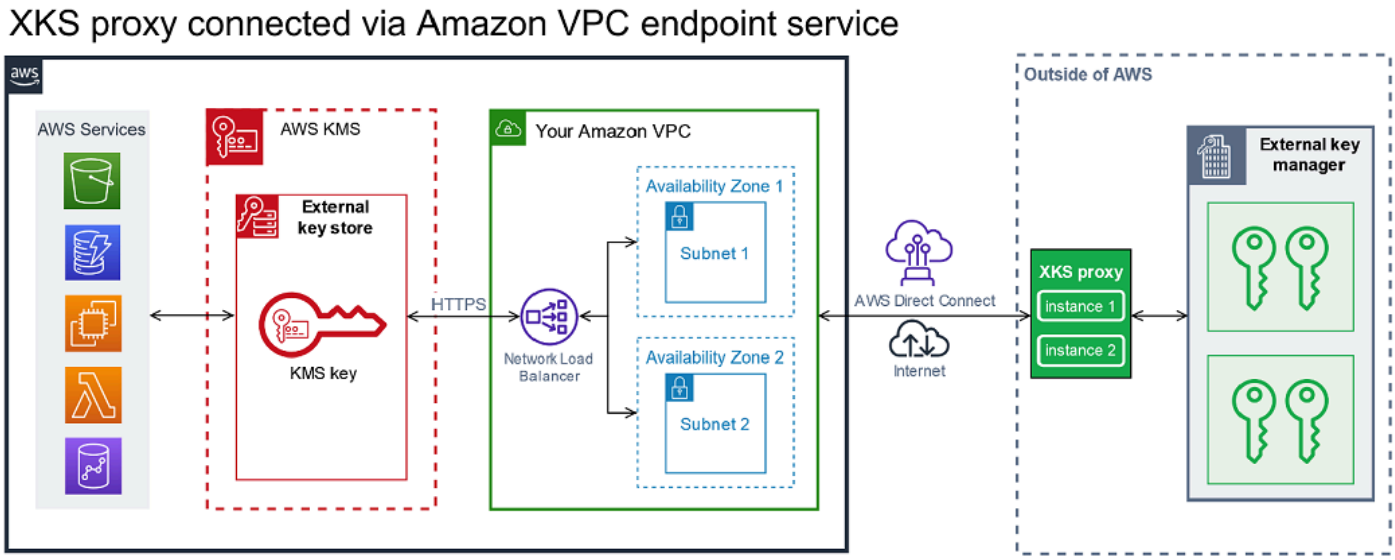
Ihr Endpunktservice kann eine der unterstützten Netzwerk–zu-Amazon VPC-Optionen für die Kommunikation nutzen, einschließlich AWS Direct Connect.

Einrichtung und Verwaltung dieser Konnektivitätsoption sind komplizierter. Sie verwendet jedoch AWS PrivateLink, wodurch AWS KMS eine private Verbindung zu Ihrer Amazon VPC und zu Ihrem externen Schlüsselspeicher-Proxy herstellen kann, ohne das öffentliche Internet zu nutzen.

Sie können Ihren externen Schlüsselspeicher-Proxy in Ihrer Amazon VPC lokalisieren.



Alternativ können Sie Ihren externen Schlüsselspeicher-Proxy außerhalb von AWS lokalisieren und Ihren Amazon VPC-Endpunktservice ausschließlich für die sichere Kommunikation mit AWS KMS verwenden.



Konfigurieren der Konnektivität eines VPC-Endpunktservice

Verwenden Sie die Informationen in diesem Abschnitt, um die AWS-Ressourcen und zugehörigen Komponenten zu erstellen und zu konfigurieren, die für einen externen Schlüsselspeicher, der die Konnektivität eines VPC-Endpunktservice verwendet, erforderlich sind. Die für diese Konnektivitätsoption aufgeführten Ressourcen sind zusätzlich zu den für alle externen

Schlüsselspeicher benötigten Ressourcen erforderlich. Nachdem Sie die erforderlichen Ressourcen erstellt und konfiguriert haben, können Sie den externen Schlüsselspeicher erstellen.

Sie können Ihren externen Schlüsselspeicher in Ihrer Amazon VPC lokalisieren oder den Proxy außerhalb von AWS lokalisieren und Ihren VPC-Endpunktservice für die Kommunikation verwenden.

Bevor Sie beginnen, vergewissern Sie sich, dass Sie einen externen Schlüsselspeicher benötigen. Die meisten Kunden können KMS-Schlüssel verwenden, die durch AWS KMS-Schlüsselmaterial gesichert sind.

> **Note**
>
> Einige der Elemente, die für die Konnektivität eines VPC-Endpunktservice erforderlich sind, sind möglicherweise in Ihrem externen Schlüsselmanager enthalten. Außerdem gelten unter Umständen für Ihre Software zusätzliche Konfigurationsanforderungen. Ziehen Sie die Dokumentation für Ihren Proxy und Schlüsselmanager zurate, bevor Sie die AWS-Ressourcen in diesem Abschnitt erstellen.

Themen

Anforderungen für die Konnektivität eines VPC-Endpunktservice

Wenn Sie sich für Ihren externen Schlüsselspeicher für die Konnektivität eines VPC-Endpunktservice entscheiden, sind die folgenden Ressourcen erforderlich.

Um die Netzwerklatenz zu minimieren, erstellen Sie Ihre AWS-Komponenten in der unterstützten AWS-Region, die Ihrem externen Schlüsselmanager am nächsten ist. Wählen Sie nach Möglichkeit eine Region mit einer Netzwerk-Round-Trip-Zeit (RTT) von maximal 35 Millisekunden.

- Eine Amazon VPC, die mit Ihrem externen Schlüsselmanager verbunden ist. Sie muss über mindestens zwei private Subnetze in zwei verschiedenen Availability Zones verfügen.

  Sie können eine vorhandene Amazon VPC für Ihren externen Schlüsselspeicher verwenden, sofern diese die Anforderungen für die Verwendung mit einem externen Schlüsselspeicher erfüllt. Mehrere externe Schlüsselspeicher können sich eine Amazon VPC teilen, aber jeder externe Schlüsselspeicher muss seinen eigenen VPC-Endpunktservice und privaten DNS-Namen haben.

- Ein Amazon VPC-Endpunktservice, der von AWS PrivateLink unterstützt wird, mit einem Network Load Balancer und einer Zielgruppe.

  Der Endpunktservice darf keine Akzeptanz verlangen. Außerdem müssen Sie AWS KMS als zulässigen Prinzipal hinzufügen. Dadurch kann AWS KMS Schnittstellenendpunkte erstellen, um mit Ihrem externen Schlüsselspeicher-Proxy zu kommunizieren.

- Ein privater DNS-Name für den VPC-Endpunktservice, der in seiner AWS-Region eindeutig ist.

  Der private DNS-Name muss eine Subdomain einer öffentlichen Domain höherer Ebene sein. Wenn der private DNS-Name beispielsweise `myproxy-private.xks.example.com` lautet, muss er eine Subdomain einer öffentlichen Domain wie `xks.example.com` oder `example.com` sein.

  Sie müssen den Besitz der DNS-Domain für den privaten DNS-Namen bestätigen.

- Ein TLS-Zertifikat für Ihren externen Schlüsselspeicher-Proxy, das von einer unterstützten öffentlichen Zertifizierungsstelle ausgestellt wurde.

  Der Subject Common Name (CN) auf dem TLS-Zertifikat muss mit dem privaten DNS-Namen übereinstimmen. Wenn der private DNS-Name beispielsweise `myproxy-private.xks.example.com` lautet, muss der CN auf dem TLS-Zertifikat `myproxy-private.xks.example.com` oder `*.xks.example.com` lauten.

Alle Anforderungen für einen externen Schlüsselspeicher finden Sie in den Informationen zum Erfüllen der Voraussetzungen.

Erstellen einer Amazon VPC und von Subnetzen

Die Konnektivität eines VPC-Endpunktservice erfordert eine Amazon VPC, die mit Ihrem externen Schlüsselmanager mit mindestens zwei privaten Subnetzen verbunden ist. Sie können eine Amazon VPC erstellen oder eine vorhandene Amazon VPC verwenden, die die Anforderungen für externe

Schlüsselspeicher erfüllt. Informationen zum Erstellen einer neuen Amazon VPC finden Sie unter Erstellen einer VPC im Amazon Virtual Private Cloud-Benutzerhandbuch.

Anforderungen für Ihre Amazon VPC

Damit Sie unter Verwendung der Konnektivität eines VPC-Endpunktservice mit externen Schlüsselspeichern arbeiten können, muss die Amazon VPC die folgenden Eigenschaften haben:

- Sie muss sich in demselben AWS-Konto und in derselben unterstützten Region wie Ihr externer Schlüsselspeicher befinden.
- Sie muss über mindestens zwei private Subnetze verfügen, die in verschiedenen Availability Zones sind.
- Der private IP-Adressbereich Ihrer Amazon VPC darf sich nicht mit dem privaten IP-Adressbereich des Rechenzentrums überschneiden, von dem Ihr externer Schlüsselmanager gehostet wird.
- Alle Komponenten müssen IPv4 verwenden.

Sie haben zahlreiche Möglichkeiten, die Amazon VPC mit Ihrem externen Schlüsselspeicher-Proxy zu verbinden. Wählen Sie eine Option aus, die Ihre Leistungs- und Sicherheitsanforderungen erfüllt. Eine Liste finden Sie unter Verbinden Ihrer VPC mit anderen Netzwerken und Verbindungsoptionen zwischen Netzwerk und Amazon VPC. Weitere Informationen finden Sie im AWS Direct Connect und im AWS Site-to-Site VPN-Benutzerhandbuch.

Erstellen einer Amazon VPC für Ihren externen Schlüsselspeicher

Verwenden Sie folgende Anweisungen zum Erstellen der Amazon VPC für Ihren externen Schlüsselspeicher. Eine Amazon VPC ist nur erforderlich, wenn Sie die Option Konnektivität eines VPC-Endpunktservice auswählen. Sie können eine vorhandene Amazon VPC verwenden, die die Anforderungen für einen externen Schlüsselspeicher erfüllt.

Folgen Sie den Anweisungen im Thema Erstellen einer VPC sowie von Subnetzen und anderen VPC-Ressourcen und verwenden Sie dabei die folgenden erforderlichen Werte. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Wert
IPv4 CIDR block (IPv4-CIDR-Block)	Geben Sie die IP-Adressen für Ihre VPC ein. Der private IP-Adressbereich Ihrer Amazon VPC darf sich nicht mit dem privaten IP-Adressbereich des

Feld	Wert
	Rechenzentrums überschneiden, von dem Ihr externer Schlüsselmanager gehostet wird.
Number of Availability Zones (AZs) (Anzahl der Availability Zones (AZs)	2 oder mehr
Number of public subnets (Anzahl der öffentlichen Subnetze)	Keine erforderlich (0)
Number of private subnets (Anzahl der privaten Subnetze)	Eines pro AZ
NAT gateways (NAT-Gate ways)	Keine erforderlich
VPC endpoints (VPC-Endp unkte)	Keine erforderlich
Enable DNS hostnames (DNS-Host namen aktiviere n)	Ja

Feld	Wert
Enable DNS resolution (DNS-Auflösung aktivieren)	Ja

Testen Sie unbedingt die VPC-Kommunikation. Wenn sich Ihr externer Schlüsselspeicher-Proxy beispielsweise nicht in Ihrer Amazon VPC befindet, erstellen Sie eine Amazon EC2-Instance in Ihrer Amazon VPC und verifizieren Sie, dass die Amazon VPC mit Ihrem externen Schlüsselspeicher-Proxy kommunizieren kann.

Verbinden der VPC mit dem externen Schlüsselmanager

Verbinden Sie die VPC mit dem Rechenzentrum, von dem Ihr externer Schlüsselmanager unter Verwendung einer der von Amazon VPC unterstützten Netzwerkkonnektivitätsoptionen gehostet wird. Stellen Sie sicher, dass die Amazon EC2-Instance in der VPC (oder der externe Schlüsselspeicher-Proxy, wenn sich dieser in der VPC befindet) mit dem Rechenzentrum und dem externen Schlüsselmanager kommunizieren kann.

Erstellen einer Zielgruppe

Bevor Sie den erforderlichen VPC-Endpunktservice erstellen, erstellen Sie die zugehörigen erforderlichen Komponenten, einen Network Load Balancer (NLB) und eine Zielgruppe. Der NLB verteilt die Anfragen an mehrere fehlerfreie Ziele, von denen jedes die Anfrage erfüllen kann. In diesem Schritt erstellen Sie eine Zielgruppe mit mindestens zwei Hosts für Ihren externen Schlüsselspeicher-Proxy und registrieren Ihre IP-Adressen bei der Zielgruppe.

Folgen Sie den Anweisungen im Thema Konfigurieren einer Zielgruppe und verwenden Sie dabei die folgenden erforderlichen Werte. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Wert
Target type (Zieltyp)	IP-Adressen

Feld	Wert
Protocol (Protokoll)	TCP
Port	443
IP address type (IP-Adresstyp)	IPv4
VPC	Wählen Sie die VPC aus, in der Sie den VPC-Endpunktservice für Ihren externen Schlüsselspeicher erstellen werden.
Health check protocol and path (Zustands prüfungs-Protokoll und -Pfad)	Ihr Zustandsprüfungs-Protokoll und Ihr Zustandsprüfungs-Pfad werden sich von der Konfiguration für Ihren externen Schlüsselspeicher-Proxy untersche iden. Weitere Informationen finden Sie in der Dokumentation für Ihren externen Schlüsselmanager oder externen Schlüsselspeicher-Proxy. Allgemeine Informationen zum Konfigurieren von Zustandsprüfungen für Ihre Zielgruppen finden Sie unter Zustandsprüfungen für Ihre Zielgruppen im Benutzerhandbuch zum Elastic Load Balancing für Network Load Balancer.
Network (Netzwerk)	Andere private IP-Adresse
IPv4 address (IPv4-Adresse)	Die privaten Adressen für Ihren externen Schlüsselspeicher-Proxy
Ports	443

Erstellen eines Network Load Balancers

Der Network Load Balancer verteilt die Netzwerkzugriffe, einschließlich Anfragen von AWS KMS an Ihren externen Schlüsselspeicher-Proxy, an die konfigurierten Ziele.

Folgen Sie den Anweisungen im Thema Konfigurieren eines Load Balancers und eines Listeners, um unter Verwendung der folgenden erforderlichen Werte einen Listener zu konfigurieren und hinzuzufügen und einen Load Balancer zu erstellen. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Wert
Scheme (Schema)	Intern
IP address type (IP-Adresstyp)	IPv4
Network mapping (Netzwerk zuordnung)	Wählen Sie die VPC aus, in der Sie den VPC-Endpunktservice für Ihren externen Schlüsselspeicher erstellen werden.
Mapping (Zuordnung)	Wählen Sie die beiden Availability Zones (mindestens zwei) aus, die Sie für Ihre VPC-Subnetze konfiguriert haben. Verifizieren Sie die Namen der Subnetze und die private IP-Adresse.
Protocol (Protokoll)	TCP
Port	443
Default action: Forward to (Standard aktion: Weiterlei ten an)	Wählen Sie die Zielgruppe für Ihren Network Load Balancer aus.

Erstellen eines VPC-Endpunktservice

Normalerweise erstellen Sie einen Endpunkt für einen Service. Wenn Sie jedoch einen VPC-Endpunktservice erstellen, sind Sie der Anbieter und AWS KMS erstellt einen Endpunkt für Ihren Service. Für einen externen Schlüsselspeicher erstellen Sie einen VPC-Endpunktservice mit dem Network Load Balancer, den Sie im vorherigen Schritt erstellt haben. Der VPC-Endpunktservice muss sich in demselben AWS-Konto und in derselben unterstützten Region wie Ihr externer Schlüsselspeicher befinden.

Mehrere externe Schlüsselspeicher können sich eine Amazon VPC teilen, aber jeder externe Schlüsselspeicher muss seinen eigenen VPC-Endpunktservice und privaten DNS-Namen haben.

Folgen Sie den Anweisungen im Thema Erstellen eines Endpunktservice, um Ihren VPC-Endpunktservice mit den folgenden erforderlichen Werten zu erstellen. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Wert
Load balancer type (Load Balancer-Typ)	Network (Netzwerk)
Available load balancers (Verfügbare Load Balancer)	Wählen Sie den Network Load Balancer aus, den Sie im vorherigen Schritt erstellt haben. Falls Ihr neuer Load Balancer nicht in der Liste aufgeführt wird, stellen Sie sicher, dass er aktiv ist. Es kann einige Minuten dauern, bis der Zustand des Load Balancers von der Bereitstellung in den aktiven Status ändert.
Acceptance required (Akzeptanz erforderlich)	Falsch. Wählen Sie dieses Kontrollkästchen ab. Verlangen Sie keine Akzeptanz. AWS KMS kann ohne manuelle Akzeptanz keine Verbindung mit dem VPC-Endpunktservice herstellen. Wenn die Akzeptanz erforderlich ist, schlägt das Erstellen des externen Schlüsselspeichers mit einer `XksProxyInvalidConfigurationException` - Ausnahme fehl.
Enable private DNS name (Privaten DNS-Namen aktivieren)	Ordnen Sie dem Service einen privaten DNS-Namen zu.
Private DNS name (Privater DNS-Name)	Geben Sie einen privaten DNS-Namen ein, der in seiner AWS-Region eindeutig ist. Der private DNS-Name muss eine Subdomain einer öffentlichen Domain höherer Ebene sein. Wenn der private DNS-Name beispielsweise `myproxy-`

Feld	Wert
	`private.xks.example.com` lautet, muss er eine Subdomain einer öffentlichen Domain wie `xks.example.com` oder `example.com` sein. Dieser private DNS-Name muss mit dem Subject Common Name (CN) im TLS-Zertifikat übereinstimmen, das auf Ihrem externen Schlüsselspeicher-Proxy konfiguriert ist. Wenn der private DNS-Name beispielsweise `myproxy-private.xks.example.com` lautet, muss der CN auf dem TLS-Zertifikat `myproxy-private.xks.example.com` oder `*.xks.example.com` lauten. Falls das Zertifikat und der private DNS-Name nicht übereinstimmen, schlagen Versuche, einen externen Schlüsselspeicher mit seinem externen Schlüssel speicher-Proxy zu verbinden, mit dem Verbindungsfehlercode `XKS_PROXY_INVALID_TLS_CONFIGURATION` fehl. Details hierzu finden Sie unter Allgemeine Konfigurationsfehler.
Supported IP address types (Unterstützte IP-Adresstypen)	IPv4

Verifizieren der Domain Ihres privaten DNS-Namens

Wenn Sie einen VPC-Endpunktservice erstellen, lautet der Verfizierungsstatus für die zugehörige Domain `pendingVerification`. Bevor Sie den VPC-Endpunktservice verwenden, um einen externen Schlüsselspeicher zu erstellen, muss dieser Status `verified` lauten. Um zu bestätigen, dass Sie der Besitzer der Domain sind, die Ihrem privaten DNS-Namen zugeordnet ist, müssen Sie einen TXT-Datensatz in einem öffentlichen DNS-Server erstellen.

Lautet beispielsweise der private DNS-Name für Ihren VPC-Endpunktservice `myproxy-private.xks.example.com`, müssen Sie einen TXT-Datensatz in einer öffentlichen Domain wie `xks.example.com` oder `example.com` erstellen (je nachdem, welche davon öffentlich sind). AWS PrivateLink sucht zuerst auf `xks.example.com` und dann auf `example.com` nach dem TXT-Datensatz.

> **Tip**
>
> Nachdem Sie einen TXT-Datensatz hinzugefügt haben, kann es einige Minuten dauern, bis sich der Wert von Domain verification status (Domain-Verifizierungsstatus) von `pendingVerification` in `verify` ändert.

Ermitteln Sie zuerst mithilfe einer der nachfolgenden Methoden, welchen Verifizierungsstatus Ihre Domain hat. Gültige Werte sind `verified`, `pendingVerification` und `failed`.

- Wählen Sie in der Amazon VPC-Konsole die Option Endpoint services (Endpunktservices) und dann Ihren Endpunktservice aus. Im Detailbereich wird Domain verification status (Domain-Verifizierungsstatus) angezeigt.
- Verwenden Sie die -DescribeVpcEndpointServiceConfigurationsOperation. Der Wert für `State` ist im Feld `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Sollte der Verifizierungsstatus nicht `verified` lauten, folgen Sie den Anweisungen im Thema Domain-Eigentumsüberprüfung, um dem DNS-Server Ihrer Domain einen TXT-Datensatz hinzuzufügen und zu verifizieren, dass der TXT-Datensatz veröffentlicht wird. Überprüfen Sie den Verifizierungsstatus danach erneut.

Sie sind nicht dazu verpflichtet, einen A-Datensatz für den Namen Ihrer privaten DNS-Domain zu erstellen. Wenn AWS KMS einen Schnittstellen-Endpunkt für Ihren VPC-Endpunktservice erstellt, erstellt AWS PrivateLink automatisch eine gehostete Zone mit dem erforderlichen A-Datensatz für den Namen der privaten Domain in der AWS KMS VPC. Für externe Schlüsselspeicher mit der Konnektivität eines VPC-Endpunktservice geschieht dies, wenn Sie Ihren externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbinden.

Autorisieren von AWS KMS, eine Verbindung mit dem VPC-Endpunktservice herzustellen

Sie müssen AWS KMS der Liste der zulässigen Prinzipale für Ihren VPC-Endpunktservice hinzufügen. Dadurch kann AWS KMS Schnittstellen-Endpunkte für Ihren VPC-Endpunktservice erstellen. Wenn AWS KMS kein zulässiger Prinzipal ist, schlagen Versuche, einen externen Schlüsselspeicher zu erstellen, mit einer `XksProxyVpcEndpointServiceNotFoundException`-Ausnahme fehl.

Folgen Sie den Anweisungen im Thema Verwalten von Berechtigungen im AWS PrivateLink-Benutzerhandbuch. Verwenden Sie den folgenden erforderlichen Wert.

Feld	Wert
ARN	`cks.kms.<region>.amazonaws.com` Beispiel: `cks.kms.us-east-1.amazonaws.com`

Weiter: Erstellen eines externen Schlüsselspeichers

Verwaltung eines externen Schlüsselspeichers

Sie können einen externen Schlüsselspeicher mithilfe der AWS KMS-Konsole oder der AWS KMS-API verwalten. Sie können einen externen Schlüsselspeicher erstellen, seine Eigenschaften anzeigen und bearbeiten, seine Leistung überwachen, ihn mit dem Proxy des externen Schlüsselspeichers verbinden und von ihm trennen sowie den externen Schlüsselspeicher löschen.

Themen

- Erstellen eines externen Schlüsselspeichers
- Bearbeiten der Eigenschaften eines externen Schlüsselspeichers
- Anzeigen eines externen Schlüsselspeichers
- Überwachung eines externen Schlüsselspeichers
- Herstellen und Trennen der Verbindung eines externen Schlüsselspeichers
- Löschen eines externen Schlüsselspeichers

Erstellen eines externen Schlüsselspeichers

Sie können in jedem AWS-Konto und jeder Region einen oder mehrere externe Schlüsselspeicher erstellen. Jeder externe Schlüsselspeicher muss mit einem externen Schlüsselmanager außerhalb von AWS und einem externen Schlüsselspeicher-Proxy (XKS-Proxy) verbunden sein, der die Kommunikation zwischen AWS KMS und Ihrem externen Schlüsselmanager vermittelt. Details hierzu finden Sie unter Planen eines externen Schlüsselspeichers. Bevor Sie beginnen, vergewissern Sie sich, dass Sie einen externen Schlüsselspeicher benötigen. Die meisten Kunden können KMS-Schlüssel verwenden, die durch AWS KMS-Schlüsselmaterial gesichert sind.

> **Tip**
>
> Einige externe Schlüsselmanager bieten eine einfachere Methode zum Erstellen eines externen Schlüsselspeichers. Weitere Informationen finden Sie in der Dokumentation zum externen Schlüsselmanager.

Vor dem Erstellen des externen Schlüsselspeichers müssen Sie einige Voraussetzungen erfüllen. Geben Sie während des Erstellungsprozesses die Eigenschaften Ihres externen Schlüsselspeichers an. Vor allem müssen Sie angeben, ob Ihr externer Schlüsselspeicher in AWS KMS einen öffentlichen Endpunkt oder einen VPC-Endpunkt-Service verwendet, um eine Verbindung zu seinem externen Schlüsselspeicher-Proxy herzustellen. Sie geben auch die Verbindungsdetails an, einschließlich des URI-Endpunkts des Proxys und des Pfads innerhalb dieses Proxy-Endpunkts, an den AWS KMS API-Anforderungen an den Proxy sendet.

- Wenn Sie eine öffentliche Endpunktverbindung verwenden, stellen Sie sicher, dass AWS KMS über eine HTTPS-Verbindung mit Ihrem Proxy über das Internet kommunizieren kann. Dazu gehört die Konfiguration von TLS auf dem externen Schlüsselspeicher-Proxy und die Sicherstellung, dass alle Firewalls zwischen AWS KMS und dem Proxy den Datenverkehr zu und von Port 443 auf dem Proxy zulassen. Beim Erstellen eines externen Schlüsselspeichers mit der Konnektivität eines öffentlichen Endpunkts testet AWS KMS die Verbindung, indem es eine Statusanfrage an den externen Schlüsselspeicher-Proxy sendet. Mit diesem Test wird überprüft, ob der Endpunkt erreichbar ist und ob der externe Schlüsselspeicher-Proxy eine mit der Anmeldeinformation für die Proxy-Authentifizierung des externen Schlüsselspeichers signierte Anforderung akzeptiert. Wenn diese Testanforderung fehlschlägt, schlägt der Vorgang zum Erstellen des externen Schlüsselspeichers fehl.
- Wenn Sie die Konnektivität eines VPC-Endpunkt-Service verwenden, stellen Sie sicher, dass der Network Load Balancer, der private DNS-Name und der VPC-Endpunkt-Service korrekt konfiguriert und betriebsbereit sind. Wenn sich der externe Schlüsselspeicher-Proxy nicht in der VPC befindet, müssen Sie sicherstellen, dass der VPC-Endpunkt-Service mit dem externen Schlüsselspeicher-Proxy kommunizieren kann. (AWS KMS testet die Konnektivität des VPC-Endpunkt-Service, wenn Sie den externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbinden.)

Weitere Überlegungen:

- AWS KMS zeichnet Amazon- CloudWatch Metriken und -Dimensionen speziell für externe Schlüsselspeicher auf. In der AWS KMS-Konsole werden für jeden externen Schlüsselspeicher

Überwachungsdiagramme angezeigt, die auf einigen dieser Metriken basieren. Es wird dringend empfohlen, diese Metriken zu verwenden, um Alarme zu erstellen, die Ihren externen Schlüsselspeicher überwachen. Diese Alarme warnen Sie vor ersten Anzeichen von Leistungs- und Betriebsproblemen, bevor sie auftreten. Anweisungen finden Sie unter Überwachung eines externen Schlüsselspeichers.

- Externe Schlüsselspeicher unterliegen Ressourcenkontingenten. Die Verwendung von KMS-Schlüsseln in einem externen Schlüsselspeicher unterliegt Anforderungskontingenten. Überprüfen Sie diese Kontingente, bevor Sie Ihre externe Schlüsselspeicher-Implementierung entwerfen.

> **Note**
>
> Überprüfen Sie Ihre Konfiguration auf zirkuläre Abhängigkeiten, die möglicherweise verhindern, dass sie funktioniert.
> Wenn Sie beispielsweise Ihren externen Schlüsselspeicher-Proxy mithilfe von AWS-Ressourcen erstellen, stellen Sie sicher, dass für den Betrieb des Proxys nicht die Verfügbarkeit eines KMS-Schlüssels in einem externen Schlüsselspeicher erforderlich ist, auf den über diesen Proxy zugegriffen wird.

Alle neuen externen Schlüsselspeicher werden in einem getrennten Zustand erstellt. Bevor Sie KMS-Schlüssel für Ihren externen Schlüsselspeicher erstellen können, müssen Sie ihn mit seinem externen Schlüsselspeicher-Proxy verbinden. Bearbeiten Sie die Einstellungen Ihres externen Schlüsselspeichers, um die Eigenschaften Ihres externen Schlüsselspeichers zu ändern.

Themen

Erfüllen der Voraussetzungen

Bevor Sie einen externen Schlüsselspeicher erstellen, müssen Sie die erforderlichen Komponenten zusammenstellen, einschließlich des externen Schlüsselmanagers, den Sie zur Unterstützung des externen Schlüsselspeichers verwenden werden, und des externen Schlüsselspeicher-Proxys, der

AWS KMS-Anforderungen in ein Format übersetzt, das Ihr externer Schlüsselmanager verstehen kann.

Die folgenden Komponenten sind für alle externen Schlüsselspeicher erforderlich. Zusätzlich zu diesen Komponenten müssen Sie die Komponenten bereitstellen, die die von Ihnen gewählte Option der externen Schlüsselspeicher-Proxy-Konnektivität unterstützen.

> **Tip**
>
> Ihr externer Schlüsselmanager enthält möglicherweise einige dieser Komponenten, oder sie sind möglicherweise für Sie konfiguriert. Weitere Informationen finden Sie in der Dokumentation zum externen Schlüsselmanager.
> Wenn Sie Ihren externen Schlüsselspeicher in der AWS KMS-Konsole erstellen, haben Sie die Möglichkeit, eine JSON-basierte Proxy-Konfigurationsdatei hochzuladen, die den Proxy-URI-Pfad und die Anmeldeinformationen für die Proxy-Authentifizierung angibt. Einige externe Schlüsselspeicher-Proxys generieren diese Datei für Sie. Einzelheiten finden Sie in der Dokumentation für Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager.

Externer Schlüsselmanager

Jeder externe Schlüsselspeicher benötigt mindestens eine externe Schlüsselmanager-Instance. Dies kann ein physisches oder virtuelles Hardwaresicherheitsmodul (HSM) oder eine Schlüsselverwaltungssoftware sein.

Sie können einen einzelnen Schlüsselmanager verwenden, wir empfehlen jedoch aus Redundanzgründen mindestens zwei verwandte Schlüsselmanager-Instances, die sich kryptografische Schlüssel teilen. Der externe Schlüsselspeicher erfordert keine ausschließliche Verwendung des externen Schlüsselmanagers. Der externe Schlüsselmanager muss jedoch in der Lage sein, die erwartete Häufigkeit von Verschlüsselungs- und Entschlüsselungsanforderungen der AWS-Services zu verarbeiten, die KMS-Schlüssel im externen Schlüsselspeicher verwenden, um Ihre Ressourcen zu schützen. Ihr externer Schlüsselmanager sollte so konfiguriert sein, dass er bis zu 1 800 Anfragen pro Sekunde verarbeitet und innerhalb des Zeitlimits von 250 Millisekunden für jede Anforderung reagiert. Wir empfehlen, den externen Schlüsselmanager in der Nähe einer AWS-Region zu platzieren, so dass die Netzwerk-Round-Trip-Zeit (RTT) maximal 35 Millisekunden beträgt.

Wenn Ihr externer Schlüsselspeicher-Proxy dies zulässt, können Sie den externen Schlüsselmanager ändern, den Sie Ihrem externen Schlüsselspeicher-Proxy zuordnen. Der neue externe

Schlüsselmanager muss jedoch ein Backup oder ein Snapshot mit demselben Schlüsselmaterial sein. Wenn der externe Schlüssel, den Sie einem KMS-Schlüssel zuordnen, für Ihren externen Schlüsselspeicher-Proxy nicht mehr verfügbar ist, kann AWS KMS den mit dem KMS-Schlüssel verschlüsselten Geheimtext nicht entschlüsseln.

Der externe Schlüsselmanager muss für den externen Schlüsselspeicher-Proxy zugänglich sein. Wenn die GetHealthStatus Antwort des Proxys meldet, dass alle externen Schlüsselmanager-Instances sind`Unavailable`, schlagen alle Versuche, einen externen Schlüsselspeicher zu erstellen, mit einem fehl`XksProxyUriUnreachableException`.

Externer Schlüsselspeicher-Proxy

Sie müssen einen externen Schlüsselspeicher-Proxy (XKS-Proxy) angeben, der den Designanforderungen in der API-Spezifikation von AWS KMS für externe Schlüsselspeicher-Proxys entspricht. Sie können einen externen Schlüsselspeicher-Proxy entwickeln oder kaufen oder einen externen Schlüsselspeicher-Proxy verwenden, der von Ihrem externen Schlüsselmanager bereitgestellt wird oder in diesen integriert ist. AWS KMS empfiehlt, Ihren externen Schlüsselspeicher-Proxy so zu konfigurieren, dass er bis zu 1 800 Anfragen pro Sekunde verarbeitet und innerhalb des Zeitlimits von 250 Millisekunden für jede Anforderung reagiert. Wir empfehlen, den externen Schlüsselmanager in der Nähe einer AWS-Region zu platzieren, so dass die Netzwerk-Round-Trip-Zeit (RTT) maximal 35 Millisekunden beträgt.

Sie können einen externen Schlüsselspeicher-Proxy für mehr als einen externen Schlüsselspeicher verwenden, aber jeder externe Schlüsselspeicher muss für seine Anforderungen einen eindeutigen URI-Endpunkt und -Pfad innerhalb des externen Schlüsselspeicher-Proxys haben.

Wenn Sie die Konnektivität eines VPC-Endpunkt-Service verwenden, können Sie Ihren externen Schlüsselspeicher-Proxy in Ihrer Amazon VPC lokalisieren, dies ist jedoch nicht erforderlich. Sie können Ihren Proxy außerhalb von AWS, z. B. in Ihrem privaten Rechenzentrum, lokalisieren und den VPC-Endpunkt-Service nur für die Kommunikation mit dem Proxy verwenden.

Anmeldeinformationen für die Proxy-Authentifizierung

Um einen externen Schlüsselspeicher zu erstellen, müssen Sie Ihre Anmeldeinformationen für die Proxy-Authentifizierung für den externen Schlüsselspeicher (`XksProxyAuthenticationCredential`) angeben.

Sie müssen Anmeldeinformationen für die Authentifizierung (`XksProxyAuthenticationCredential`) für AWS KMS auf Ihrem externen Schlüsselspeicher-

Proxy einrichten. AWS KMS authentifiziert sich bei Ihrem Proxy, indem es seine Anfragen mit dem Prozess Signature Version 4 (SigV4) mit den Anmeldeinformationen für die Authentifizierung des externen Schlüsselspeicher-Proxys signiert. Sie geben die Anmeldeinformationen für die Authentifizierung an, wenn Sie Ihren externen Schlüsselspeicher erstellen, und Sie können sie jederzeit ändern. Wenn Ihr Proxy Ihre Anmeldeinformationen rotiert, müssen Sie die Anmeldeinformationswerte für Ihren externen Schlüsselspeicher aktualisieren.

Die Anmeldeinformationen für die Proxy-Authentifizierung bestehen aus zwei Teilen. Sie müssen beide Teile für Ihren externen Schlüsselspeicher bereitstellen.

- Zugriffsschlüssel-ID: Identifiziert den geheimen Zugriffsschlüssel. Sie können diese ID in Klartext angeben.
- Geheimer Zugriffsschlüssel: Der geheime Teil der Anmeldeinformation. AWS KMS verschlüsselt den geheimen Zugriffsschlüssel in der Anmeldeinformation, bevor er gespeichert wird.

Die SigV4-Anmeldeinformation, die AWS KMS zum Signieren von Anfragen an den externen Schlüsselspeicher-Proxy verwendet, stehen in keinem Zusammenhang mit SigV4-Anmeldeinformationen, die mit den AWS Identity and Access Management-Prinzipalen in Ihren AWS-Konten verknüpft sind. Verwenden Sie keine IAM-SigV4-Anmeldeinformationen für Ihren externen Schlüsselspeicher-Proxy wieder.

Proxy-Konnektivität

Sie müssen die Proxy-Konnektivitätsoption des externen Schlüsselspeichers angeben (`XksProxyConnectivity`), um einen externen Schlüsselspeicher zu erstellen.

AWS KMS kann durch die Verwendung eines öffentlichen Endpunkts oder eines Amazon Virtual Private Cloud (Amazon VPC)-Endpunkt-Service mit Ihrem externen Schlüsselspeicher-Proxy kommunizieren. Ein öffentlicher Endpunkt ist zwar einfacher zu konfigurieren und zu verwalten, erfüllt jedoch möglicherweise nicht die Sicherheitsanforderungen für jede Installation. Wenn Sie sich für die Option der Konnektivität des Amazon VPC-Endpunkt-Service entscheiden, müssen Sie die erforderlichen Komponenten erstellen und verwalten, einschließlich einer Amazon VPC mit mindestens zwei Subnetzen in zwei verschiedenen Availability Zones, einem VPC-Endpunkt-Service mit einem Network Load Balancer und einer Zielgruppe sowie einem privaten DNS-Namen für den VPC-Endpunkt-Service.

Für Ihren externen Schlüsselspeicher können Sie die Proxy-Konnektivitätsoption ändern. Sie müssen jedoch sicherstellen, dass das mit den KMS-Schlüsseln verbundene Schlüsselmaterial in Ihrem

externen Schlüsselspeicher weiterhin verfügbar ist. Andernfalls kann AWS KMS keinen mit diesen KMS-Schlüsseln verschlüsselten Geheimtext entschlüsseln.

Hilfe bei der Entscheidung, welche Proxy-Konnektivitätsoption für Ihren externen Schlüsselspeicher am besten geeignet ist, finden Sie unter Auswählen einer Proxy-Konnektivitätsoption. Hilfe beim Erstellen und Konfigurieren der Konnektivität eines VPC-Endpunkt-Service finden Sie unter Konfigurieren der Konnektivität eines VPC-Endpunktservice.

Proxy-URI-Endpunkt

Um einen externen Schlüsselspeicher zu erstellen, müssen Sie den Endpunkt (`XksProxyUriEndpoint`) angeben, den AWS KMS verwendet, um Anforderungen an den externen Schlüsselspeicher-Proxy zu senden.

Das Protokoll muss HTTPS sein. AWS KMS kommuniziert über Port 443. Geben Sie den Port nicht im Wert des Proxy-URI-Endpunkts an.

- Konnektivität eines öffentlichen Endpunkts: Geben Sie den öffentlich verfügbaren Endpunkt für Ihren externen Schlüsselspeicher-Proxy an. Dieser Endpunkt muss erreichbar sein, bevor Sie Ihren externen Schlüsselspeicher erstellen.
- Konnektivität eines VPC-Endpunkt-Service: Geben Sie `https://` gefolgt von dem privaten DNS-Namen des VPC-Endpunkt-Service an.

Das TLS-Serverzertifikat, das auf dem externen Schlüsselspeicher-Proxy konfiguriert ist, muss mit dem Domainnamen im URI-Endpunkt des externen Schlüsselspeicher-Proxy übereinstimmen und von einer Zertifizierungsstelle ausgestellt sein, die für externe Schlüsselspeicher unterstützt wird. Eine Liste finden Sie unter Vertrauenswürdige Zertifizierungsstellen. Ihre Zertifizierungsstelle verlangt einen Nachweis über den Besitz der Domain, bevor sie das TLS-Zertifikat ausstellt.

Der Subject Common Name (CN) auf dem TLS-Zertifikat muss mit dem privaten DNS-Namen übereinstimmen. Wenn der private DNS-Name beispielsweise `myproxy-private.xks.example.com` lautet, muss der CN auf dem TLS-Zertifikat `myproxy-private.xks.example.com` oder `*.xks.example.com` lauten.

Sie können Ihren Proxy-URI-Endpunkt ändern, aber stellen Sie sicher, dass der externe Schlüsselspeicher-Proxy Zugriff auf das Schlüsselmaterial hat, das mit den KMS-Schlüsseln in Ihrem externen Schlüsselspeicher verbunden ist. Andernfalls kann AWS KMS keinen mit diesen KMS-Schlüsseln verschlüsselten Geheimtext entschlüsseln.

Anforderungen an die Eindeutigkeit

- Der kombinierte Wert für den Proxy-URI-Endpunkt (`XksProxyUriEndpoint`) und den Proxy-URI-Pfad (`XksProxyUriPath`) muss in dem AWS-Konto und der Region eindeutig sein.
- Externe Schlüsselspeicher mit der Konnektivität eines öffentlichen Endpunkts können denselben Proxy-URI-Endpunkt verwenden, sofern sie unterschiedliche Proxy-URI-Pfadwerte haben.
- Ein externer Schlüsselspeicher mit der Konnektivität eines öffentlichen Endpunkts kann nicht denselben Wert des Proxy-URI-Endpunkts verwenden wie ein externer Schlüsselspeicher mit der Konnektivität eines VPC-Endpunkt-Service in derselben AWS-Region, selbst wenn sich die Schlüsselspeicher in verschiedenen AWS-Konten befinden.
- Jeder externe Schlüsselspeicher mit der Konnektivität eines VPC-Endpunkts muss seinen eigenen privaten DNS-Namen haben. Der Proxy-URI-Endpunkt (privater DNS-Name) muss in dem AWS-Konto und der Region eindeutig sein.

Proxy-URI-Pfad

Um einen externen Schlüsselspeicher zu erstellen, müssen Sie in Ihrem externen Schlüsselspeicher-Proxy den Basispfad zu den erforderlichen Proxy-APIs angeben. Der Wert muss mit `/` beginnen und mit /kms/xks/v1 enden, wobei `v1` die Version der AWS KMS-API für den externen Schlüsselspeicher-Proxy darstellt. Dieser Pfad kann ein optionales Präfix zwischen den erforderlichen Elementen enthalten, z. B. `/example-prefix/kms/xks/v1`. Diesen Wert finden Sie in der Dokumentation für Ihren externen Schlüsselspeicher-Proxy.

AWS KMS sendet Proxy-Anforderungen an die Adresse, die durch die Verkettung von Proxy-URI-Endpunkt und Proxy-URI-Pfad angegeben ist. Wenn der Proxy-URI-Endpunkt beispielsweise `https://myproxy.xks.example.com` und der Proxy-URI-Pfad `/kms/xks/v1` lautet, sendet AWS KMS seine Proxy-API-Anforderungen an `https://myproxy.xks.example.com/kms/xks/v1`.

Sie können Ihren Proxy-URI-Pfad ändern, aber stellen Sie sicher, dass der externe Schlüsselspeicher-Proxy Zugriff auf das Schlüsselmaterial hat, das mit den KMS-Schlüsseln in Ihrem externen Schlüsselspeicher verbunden ist. Andernfalls kann AWS KMS keinen mit diesen KMS-Schlüsseln verschlüsselten Geheimtext entschlüsseln.

Anforderungen an die Eindeutigkeit

- Der kombinierte Wert für den Proxy-URI-Endpunkt (`XksProxyUriEndpoint`) und den Proxy-URI-Pfad (`XksProxyUriPath`) muss in dem AWS-Konto und der Region eindeutig sein.

VPC-Endpunktservice

Gibt den Namen des Amazon VPC-Endpunkt-Services an, der für die Kommunikation mit dem externen Schlüsselspeicher-Proxy verwendet wird. Diese Komponente ist nur für externe Schlüsselspeicher erforderlich, die die Konnektivität eines VPC-Endpunkt-Service verwenden. Hilfe zum Einrichten und Konfigurieren Ihres VPC-Endpunkt-Services für einen externen Schlüsselspeicher finden Sie unter Konfigurieren der Konnektivität eines VPC-Endpunktservice.

Der VPC-Endpunkt-Service muss über die folgenden Eigenschaften verfügen:

- Der VPC-Endpunkt-Service muss sich in demselben AWS-Konto und derselben Region befinden wie der externe Schlüsselspeicher.
- Er muss über einen Network Load Balancer (NLB) verfügen, der mit mindestens zwei Subnetzen verbunden ist, die jeweils in einer anderen Availability Zone liegen.
- Die Liste der zulässigen Prinzipale für den VPC-Endpunkt-Service muss den AWS KMS-Service-Prinzipal für die Region enthalten: `cks.kms.<region>.amazonaws.com`, z. B. `cks.kms.us-east-1.amazonaws.com`.
- Er darf keine Annahme von Verbindungsanforderungen verlangen.
- Er muss einen privaten DNS-Namen innerhalb einer öffentlichen Domain höherer Ebene haben. Sie könnten beispielsweise den privaten DNS-Namen „myproxy-private.xks.example.com“ in der öffentlichen Domain `xks.example.com` haben.

  Der private DNS-Name für einen externen Schlüsselspeicher mit der Konnektivität eines VPC-Endpunkt-Service muss in seiner AWS-Region eindeutig sein.
- Der Domain-Verifizierungsstatus der privaten DNS-Namensdomain muss `verified` lauten.
- Das auf dem externen Schlüsselspeicher-Proxy konfigurierte TLS-Serverzertifikat muss den privaten DNS-Hostnamen angeben, unter dem der Endpunkt erreichbar ist.

Anforderungen an die Eindeutigkeit

- Externe Schlüsselspeicher mit der Konnektivität eines VPC-Endpunkts können sich eine `Amazon VPC` teilen, aber jeder externe Schlüsselspeicher muss seinen eigenen VPC-Endpunkt-Service und privaten DNS-Namen haben.

Proxy-Konfigurationsdatei

Eine Proxy-Konfigurationsdatei ist eine optionale JSON-basierte Datei, die Werte für den Proxy-URI-Pfad und die Eigenschaften der Proxy-Authentifizierungsanmeldeinformation Ihres externen Schlüsselspeichers enthält. Beim Erstellen oder Bearbeiten eines externen Schlüsselspeichers in der AWS KMS-Konsole können Sie eine Proxy-Konfigurationsdatei hochladen, um Konfigurationswerte für Ihren externen Schlüsselspeicher bereitzustellen. Durch die Verwendung dieser Datei werden Tipp- und Einfügefehler verhindert und es wird sichergestellt, dass die Werte in Ihrem externen Schlüsselspeicher mit den Werten in Ihrem externen Schlüsselspeicher-Proxy übereinstimmen.

Die Proxy-Konfigurationsdateien werden vom externen Schlüsselspeicher-Proxy generiert. Ob Ihr externer Schlüsselspeicher-Proxy eine Proxy-Konfigurationsdatei anbietet, können Sie der Dokumentation Ihres externen Schlüsselspeicher-Proxys entnehmen.

Im Folgenden finden Sie ein Beispiel für eine gültige Proxy-Konfigurationsdatei mit fiktiven Werten.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
  }
}
```

Sie können eine Proxy-Konfigurationsdatei nur hochladen, wenn Sie einen externen Schlüsselspeicher in der AWS KMS-Konsole erstellen oder bearbeiten. Sie können sie nicht mit den UpdateCustomKeyStore Operationen CreateCustomKeyStore oder verwenden, aber Sie können die Werte in der Proxy-Konfigurationsdatei verwenden, um sicherzustellen, dass Ihre Parameterwerte korrekt sind.

Erstellen eines externen Schlüsselspeichers (Konsole)

Bevor Sie einen externen Schlüsselspeicher erstellen, lesen Sie den Planen eines externen Schlüsselspeichers, wählen Sie Ihren Proxy-Konnektivitätstyp aus und stellen Sie sicher, dass Sie alle erforderlichen Komponenten erstellt und konfiguriert haben. Wenn Sie Hilfe bei der Suche nach einem der erforderlichen Werte benötigen, ziehen Sie die Dokumentation für Ihren externen Schlüsselspeicher-Proxy oder Ihre Schlüsselverwaltungssoftware zu Rate.

> **Note**
>
> Wenn Sie einen externen Schlüsselspeicher in der AWS Management Console erstellen, können Sie eine JSON-basierte Proxy-Konfigurationsdatei mit Werten für den Proxy-URI-Pfad und die Proxy-Authentifizierungsanmeldeinformation hochladen. Einige Proxys generieren diese Datei für Sie. Sie ist nicht erforderlich.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Wählen Sie Create external key store (Erstellen eines externen Schlüsselspeichers) aus.
5. Geben Sie einen Anzeigenamen für den externen Schlüsselspeicher ein. Der Name muss unter allen externen Schlüsselspeichern in Ihrem Konto eindeutig sein.

   > **Important**
   >
   > Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

6. Wählen Sie Ihren Proxy-Konnektivitätstyp aus.

   Die von Ihnen gewählte Proxy-Konnektivität bestimmt die für Ihren externen Schlüsselspeicher-Proxy erforderlichen Komponenten. Hilfe bei dieser Auswahl finden Sie unter Auswählen einer Proxy-Konnektivitätsoption.
7. Wählen Sie den Namen des VPC-Endpunkt-Service für diesen externen Schlüsselspeicher aus oder geben Sie ihn ein. Dieser Schritt wird nur angezeigt, wenn der Proxy-Konnektivitätstyp des externen Schlüsselspeichers VPC-Endpunkt-Service ist.

   Der VPC-Endpunkt-Service und seine VPCs müssen die Anforderungen für einen externen Schlüsselspeicher erfüllen. Details hierzu finden Sie unter the section called “Erfüllen der Voraussetzungen”.

8. Geben Sie Ihren Proxy-URI-Endpunkt ein. Das Protokoll muss HTTPS sein. AWS KMS kommuniziert über Port 443. Geben Sie den Port nicht im Wert des Proxy-URI-Endpunkts an.

   Wenn AWS KMS den VPC-Endpunkt-Service erkennt, den Sie im vorherigen Schritt angegeben haben, wird dieses Feld für Sie ausgefüllt.

   Für die Konnektivität eines öffentlichen Endpunkts geben Sie einen öffentlich verfügbaren Endpunkt-URI ein. Für die Konnektivität des VPC-Endpunkts geben Sie `https://` gefolgt vom privaten DNS-Namen des VPC-Endpunkt-Service ein.

9. Um die Werte für das Präfix des Proxy-URI-Pfads und die Proxy-Authentifizierungsanmeldeinformation einzugeben, laden Sie eine Proxy-Konfigurationsdatei hoch oder geben Sie die Werte manuell ein.

   - Wenn Sie über eine optionale Proxy-Konfigurationsdatei verfügen, die Werte für den Proxy-URI-Pfad und die Proxy-Authentifizierungsanmeldeinformation enthält, wählen Sie Upload configuration file (Konfigurationsdatei hochladen) aus. Folgen Sie den Schritten zum Hochladen der Datei.

     Wenn die Datei hochgeladen ist, zeigt die Konsole die Werte aus der Datei in bearbeitbaren Feldern an. Sie können die Werte jetzt ändern oder diese Werte bearbeiten, nachdem der externe Schlüsselspeicher erstellt wurde.

     Wählen Sie Show secret access key (Geheimen Zugriffsschlüssel anzeigen) aus, um den Wert des geheimen Zugriffsschlüssels anzuzeigen.

   - Wenn Sie über keine Proxy-Konfigurationsdatei verfügen, können Sie den Proxy-URI-Pfad und die Werte für die Proxy-Authentifizierungsanmeldeinformation manuell eingeben.

     a. Wenn Sie nicht über eine Proxy-Konfigurationsdatei verfügen, können Sie den Proxy-URI manuell eingeben. Die Konsole liefert den erforderlichen /kms/xks/v1-Wert.

        Wenn Ihr Proxy-URI-Pfad ein optionales Präfix enthält, wie z. B. das `example-prefix` in `/example-prefix/kms/xks/v1`, geben Sie das Präfix in das Feld Proxy URI path prefix (Proxy-URI-Pfadpräfix) ein. Andernfalls lassen Sie das Feld leer.

     b. Wenn Sie nicht über eine Proxy-Konfigurationsdatei verfügen, können Sie die Proxy-Authentifizierungsanmeldeinformation manuell eingeben. Sowohl die Zugriffsschlüssel-ID als auch der geheime Zugriffsschlüssel sind erforderlich.

        - Geben Sie in Proxy credential: Access key ID (Proxy-Anmeldeinformationen: Zugriffsschlüssel-ID) die Zugriffsschlüssel-ID der Proxy-

Authentifizierungsanmeldeinformation ein. Die Zugriffsschlüssel-ID identifiziert den geheimen Zugriffsschlüssel.

- Geben Sie in Proxy credential: Secret access key (Proxy-Anmeldeinformationen: Geheimer Zugriffsschlüssel) den geheimen Zugriffsschlüssel der Proxy-Authentifizierungsanmeldeinformation ein.

Wählen Sie Show secret access key (Geheimen Zugriffsschlüssel anzeigen) aus, um den Wert des geheimen Zugriffsschlüssels anzuzeigen.

Mit diesem Verfahren wird die Authentifizierungsanmeldeinformation, die Sie auf Ihrem externen Schlüsselspeicher-Proxy eingerichtet haben, weder festgelegt noch geändert. Es verknüpft diese Werte lediglich mit Ihrem externen Schlüsselspeicher. Informationen zum Einrichten, Ändern und Rotieren der Proxy-Authentifizierungsanmeldeinformation finden Sie in der Dokumentation zu Ihrem externen Schlüsselspeicher-Proxy oder Ihrer Schlüsselverwaltungssoftware.

Wenn sich die Proxy-Authentifizierungsanmeldeinformation ändert, bearbeiten Sie die Einstellungen für die Anmeldeinformation für Ihren externen Schlüsselspeicher.

10. Wählen Sie Create external key store (Erstellen eines externen Schlüsselspeichers) aus.

Wenn der Vorgang erfolgreich war, wird der neue externe Schlüsselspeicher in der Liste der externen Schlüsselspeicher im Konto und in der Region angezeigt. Bei nicht erfolgreicher Ausführung wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt CreateKey Fehler für den externen Schlüssel.

Nächster Schritt: Neue externe Schlüsselspeicher werden nicht automatisch verbunden. Bevor Sie AWS KMS keys in Ihrem externen Schlüsselspeicher erstellen können, müssen Sie den externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbinden.

Erstellen eines externen Schlüsselspeichers (API)

Sie können die -CreateCustomKeyStoreOperation verwenden, um einen neuen externen Schlüsselspeicher zu erstellen. Hilfe bei der Suche nach den Werten für die erforderlichen Parameter finden Sie in der Dokumentation zu Ihrem externen Schlüsselspeicher-Proxy oder Ihrer Schlüsselverwaltungssoftware.

> **Tip**
>
> Sie können keine Proxy-Konfigurationsdatei hochladen, wenn Sie den `CreateCustomKeyStore`-Vorgang verwenden. Sie können jedoch die Werte in der Proxy-Konfigurationsdatei verwenden, um sicherzustellen, dass Ihre Parameterwerte korrekt sind.

Zum Erstellen eines externen Schlüsselspeichers sind für den `CreateCustomKeyStore`-Vorgang die folgenden Parameterwerte erforderlich:

- `CustomKeyStoreName`: Ein Anzeigename für den externen Schlüsselspeicher, der im Konto eindeutig ist.

  > **Important**
  >
  > Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokollen und anderen Ausgaben im Klartext angezeigt werden.

- `CustomKeyStoreType`: Geben Sie `EXTERNAL_KEY_STORE` an.
- `XksProxyConnectivity`: Geben Sie `PUBLIC_ENDPOINT` oder `VPC_ENDPOINT_SERVICE` an.
- `XksProxyAuthenticationCredential`: Geben Sie sowohl die Zugriffsschlüssel-ID als auch den geheimen Zugriffsschlüssel an.
- `XksProxyUriEndpoint`: Der Endpunkt, den AWS KMS für die Kommunikation mit Ihrem externen Schlüsselspeicher-Proxy verwendet.
- `XksProxyUriPath`: Der Pfad innerhalb des Proxys zu den Proxy-APIs.
- `XksProxyVpcEndpointServiceName`: Nur erforderlich, wenn Ihr `XksProxyConnectivity`-Wert `VPC_ENDPOINT_SERVICE` ist.

> **Note**
>
> Wenn Sie AWS CLI Version 1.0 verwenden, führen Sie den folgenden Befehl aus, bevor Sie einen Parameter mit einem HTTP- oder HTTPS-Wert angeben, wie beispielsweise den Parameter `XksProxyUriEndpoint`.
>
> ```
> aws configure set cli_follow_urlparam false
> ```

Andernfalls ersetzt AWS CLI Version 1.0 den Parameterwert durch den unter dieser URI-Adresse gefundenen Inhalt und verursacht den folgenden Fehler:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

In den folgenden Beispielen werden fiktive Werte verwendet. Bevor Sie den Befehl ausführen, ersetzen Sie sie durch gültige Werte für Ihren externen Schlüsselspeicher.

Erstellen Sie einen externen Schlüsselspeicher mit der Konnektivität eines öffentlichen Endpunkts.

```
$ 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>
```

Erstellen Sie einen externen Schlüsselspeicher mit der Konnektivität eines VPC-Endpunkt-Service.

```
$ 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>
```

Bei einer erfolgreichen Produktion gibt `CreateCustomKeyStore` die ID des benutzerdefinierten Schlüsselspeichers zurück, wie in der folgenden Beispielantwort dargestellt.

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Wenn die Produktion fehlschlägt, korrigieren Sie den in der Ausnahme angegebenen Fehler und versuchen Sie es erneut. Weitere Informationen finden Sie unter Fehlerbehebung bei externen Schlüsselspeichern.

Nächster Schritt: Um den externen Schlüsselspeicher zu verwenden, verbinden Sie ihn mit dem externen Schlüsselspeicher-Proxy.

Bearbeiten der Eigenschaften eines externen Schlüsselspeichers

Sie können ausgewählte Eigenschaften eines vorhandenen externen Schlüsselspeichers bearbeiten.

Sie können einige Eigenschaften bearbeiten, während der externe Schlüsselspeicher verbunden oder getrennt ist. Für andere Eigenschaften müssen Sie zuerst Ihren externen Schlüsselspeicher vom Proxy des externen Schlüsselspeichers trennen. Der Verbindungsstatus des externen Schlüsselspeichers muss `DISCONNECTED` sein. Während ein externer Schlüsselspeicher getrennt ist, können Sie den Schlüsselspeicher und seine KMS-Schlüssel verwalten, aber Sie können keine KMS-Schlüssel im externen Schlüsselspeicher erstellen oder verwenden. Um den Verbindungsstatus Ihres externen Schlüsselspeichers zu ermitteln, verwenden Sie die -DescribeCustomKeyStoresOperation oder sehen Sie sich den Abschnitt Allgemeine Konfiguration auf der Detailseite für den externen Schlüsselspeicher an.

Bevor Sie die Eigenschaften aktualisieren, die Ihr externer Schlüsselspeicher hat, AWS KMS sendet eine GetHealthStatus Anforderung an den externen Schlüsselspeicher-Proxy unter Verwendung der neuen Werte. Wenn die Anforderung erfolgreich ist, bedeutet dies, dass Sie eine Verbindung zu einem Proxy des externen Schlüsselspeichers herstellen und sich mit den aktualisierten Eigenschaftswerten authentifizieren können. Schlägt die Anforderung fehl, schlägt der Bearbeitungsvorgang mit einer Ausnahme fehl, die den Fehler identifiziert.

Wenn der Bearbeitungsvorgang abgeschlossen ist, werden die aktualisierten Eigenschaftswerte für Ihren externen Schlüsselspeicher in der AWS KMS-Konsole und in der `DescribeCustomKeyStores`-Antwort angezeigt. Es kann jedoch bis zu fünf Minuten dauern, bis die Änderungen vollständig wirksam werden.

Wenn Sie Ihren externen Schlüsselspeicher in der AWS KMS-Konsole bearbeiten, haben Sie die Möglichkeit, eine JSON-basierte Proxy-Konfigurationsdatei hochzuladen, die den Proxy-URI-Pfad und die Anmeldeinformationen für die Proxy-Authentifizierung angibt. Einige externe Schlüsselspeicher-Proxys generieren diese Datei für Sie. Einzelheiten finden Sie in der Dokumentation für Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager.

> **Warning**
>
> Die aktualisierten Eigenschaftswerte müssen Ihren externen Schlüsselspeicher mit einem Proxy für denselben externen Schlüsselmanager wie die vorherigen Werte oder für eine Sicherung oder einen Snapshot des externen Schlüsselmanagers mit denselben kryptografischen Schlüsseln verbinden. Wenn Ihr externer Schlüsselspeicher dauerhaft den Zugriff auf die externen Schlüssel verliert, die mit seinen KMS-Schlüsseln verknüpft sind, kann der mit diesen externen Schlüsseln verschlüsselte Geheimtext nicht wiederhergestellt werden. Insbesondere die Änderung der Proxy-Konnektivität eines externen Schlüsselspeichers kann verhindern, dass AWS KMS auf Ihre externen Schlüssel zugreift.

> **Tip**
>
> Einige externe Schlüsselmanager bieten eine einfachere Methode zur Bearbeitung der Eigenschaften externer Schlüsselspeicher. Weitere Informationen finden Sie in der Dokumentation zum externen Schlüsselmanager.

Sie können die folgenden Eigenschaften eines externen Schlüsselspeichers ändern:

Bearbeitbare Eigenschaften eines externen Schlüsselspeichers	Beliebiger Verbindungsstatus	Status „Disconnected“ (Verbindung getrennt) erforderlich
Name des benutzerdefinierten Schlüsselspeichers Ein erforderlicher Anzeigename für einen benutzerdefinierten Schlüsselspeicher. **Important** Geben Sie keine vertraulichen oder sensiblen Informationen in dieses Feld ein. Dieses Feld kann in CloudTrail Protokoll	✓	

Bearbeitbare Eigenschaften eines externen Schlüsselspeichers	Beliebiger Verbindungsstatus	Status „Disconnected“ (Verbindung getrennt) erforderlich
en und anderen Ausgaben im Klartext angezeigt werden.		
Anmeldeinformationen für die Proxy-Authentifizierung (XksProxyAuthenticationCredential) (Sie müssen sowohl die Zugriffsschlüssel-ID als auch den geheimen Zugriffsschlüssel angeben, auch wenn Sie nur ein Element ändern.)	✓	
Proxy-URI-Pfad (XksProxyUriPath)	✓	
Proxy-Konnektivität (XksProxyConnectivity) (Sie müssen auch den Proxy-URI-Endpunkt aktualisieren. Wenn Sie zur Konnektivität eines VPC-Endpunkt-Service wechseln, müssen Sie einen Namen für den Proxy-VPC-Endpunkt-Service angeben.)		✓
Proxy-URI-Endpunkt (XksProxyUriEndpoint) Wenn Sie den Proxy-Endpunkt-URI ändern, müssen Sie möglicherweise auch das zugehörige TLS-Zertifikat ändern.		✓
Name des Proxy-VPC-Endpunktservice (XksProxyVpcEndpointServiceName) (Dieses Feld ist für die Konnektivität eines VPC-Endpunkt-Service erforderlich)		✓

Themen

Bearbeiten eines externen Schlüsselspeichers (Konsole)

Wenn Sie einen Schlüsselspeicher bearbeiten, können Sie die bearbeitbaren Werte beliebig ändern. Einige Änderungen erfordern, dass der externe Schlüsselspeicher von seinem externen Schlüsselspeicher-Proxy getrennt wird.

Wenn Sie den Proxy-URI-Pfad oder die Anmeldeinformation für die Proxy-Authentifizierung bearbeiten, können Sie die neuen Werte eingeben oder eine Proxykonfigurationsdatei für den externen Schlüsselspeicher hochladen, die die neuen Werte enthält.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Wählen Sie die Zeile des externen Schlüsselspeichers aus, den Sie bearbeiten möchten.
5. Falls erforderlich, trennen Sie den externen Schlüsselspeicher von seinem externen Schlüsselspeicher-Proxy. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Disconnect (Verbindung trennen) aus.
6. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Edit (Bearbeiten) aus.
7. Ändern Sie eine oder mehrere der bearbeitbaren Eigenschaften des externen Schlüsselspeichers. Sie können auch eine Proxy-Konfigurationsdatei für den externen Schlüsselspeicher mit Werten für den Proxy-URI-Pfad und der Anmeldeinformation für die Proxy-Authentifizierung hochladen. Sie können eine Proxy-Konfigurationsdatei auch dann verwenden, wenn sich einige in der Datei angegebene Werte nicht geändert haben.
8. Wählen Sie Update external key store (Aktualisieren des externen Schlüsselspeichers).
9. Überprüfen Sie die Warnung, und wenn Sie fortfahren möchten, bestätigen Sie die Warnung und wählen Sie dann Update external key store (Aktualisieren des externen Schlüsselspeichers).

   Wenn der Vorgang erfolgreich ist, wird eine Meldung mit einer Beschreibung der von Ihnen bearbeiteten Eigenschaften angezeigt. Wenn der Vorgang nicht erfolgreich ist, wird Ihnen eine

Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt.

10. Falls erforderlich, verbinden Sie den externen Schlüsselspeicher wieder. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Connect (Verbinden) aus.

    Sie können die Verbindung des externen Schlüsselspeichers getrennt lassen. Solange die Verbindung getrennt ist, können Sie jedoch keine KMS-Schlüssel im externen Schlüsselspeicher erstellen oder die KMS-Schlüssel im externen Schlüsselspeicher für kryptografische Vorgänge verwenden.

Bearbeiten eines externen Schlüsselspeichers (API)

Um die Eigenschaften eines externen Schlüsselspeichers zu ändern, verwenden Sie die -UpdateCustomKeyStoreOperation. Sie können mehrere Eigenschaften eines externen Schlüsselspeichers mit demselben Vorgang ändern. Wenn die Produktion erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück.

Verwenden Sie den `CustomKeyStoreId`-Parameter, um den externen Schlüsselspeicher zu identifizieren. Verwenden Sie die anderen Parameter, um die Eigenschaften zu ändern. Sie können für den `UpdateCustomKeyStore`-Vorgang keine Proxy-Konfigurationsdatei verwenden. Die Proxy-Konfigurationsdatei wird nur von der AWS KMS-Konsole unterstützt. Sie können aber die Proxy-Konfigurationsdatei verwenden, um die richtigen Parameterwerte für Ihren externen Schlüsselspeicher-Proxy zu ermitteln.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Bevor Sie beginnen, trennen Sie ggf. die Verbindung des externen Schlüsselspeichers mit dem Proxy des externen Schlüsselspeichers. Nach der Aktualisierung können Sie den externen Schlüsselspeicher bei Bedarf wieder mit dem Proxy des externen Schlüsselspeichers verbinden. Sie können den externen Schlüsselspeicher im getrennten Zustand belassen, müssen ihn aber wieder verbinden, bevor Sie neue KMS-Schlüssel im Schlüsselspeicher erstellen oder vorhandene KMS-Schlüssel im Schlüsselspeicher für kryptografische Vorgänge verwenden können.

> **Note**
>
> Wenn Sie AWS CLI Version 1.0 verwenden, führen Sie den folgenden Befehl aus, bevor Sie einen Parameter mit einem HTTP- oder HTTPS-Wert angeben, wie beispielsweise den Parameter `XksProxyUriEndpoint`.

```
aws configure set cli_follow_urlparam false
```

Andernfalls ersetzt AWS CLI Version 1.0 den Parameterwert durch den unter dieser URI-Adresse gefundenen Inhalt und verursacht den folgenden Fehler:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

Ändern des Namens des externen Schlüsselspeichers

Im ersten Beispiel wird die -UpdateCustomKeyStoreOperation verwendet, um den Anzeigenamen des externen Schlüsselspeichers in zu ändern`XksKeyStore`. Der Befehl verwendet den Parameter `CustomKeyStoreId`, um den benutzerdefinierten Schlüsselspeicher zu identifizieren, und den Parameter `CustomKeyStoreName`, um den neuen Namen für den benutzerdefinierten Schlüsselspeicher anzugeben. Ersetzen Sie alle Beispielwerte durch tatsächliche Werte für Ihren externen Schlüsselspeicher.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name XksKeyStore
```

Ändern der Anmeldeinformation für die Proxy-Authentifizierung

Im folgenden Beispiel wird die Anmeldeinformation für die Proxy-Authentifizierung aktualisiert, die AWS KMS für die Authentifizierung beim Proxy des externen Schlüsselspeichers verwendet. Sie können einen Befehl wie diesen verwenden, um die Anmeldeinformation zu aktualisieren, wenn sie auf Ihrem Proxy rotiert werden.

Aktualisieren Sie zuerst die Anmeldeinformation auf Ihrem externen Schlüsselspeicher-Proxy. Verwenden Sie dann dieses Feature, um die Änderung an AWS KMS zu melden. (Ihr Proxy wird kurzzeitig sowohl die alte als auch die neue Anmeldeinformation unterstützen, damit Sie Zeit haben, Ihre Anmeldeinformation in AWS KMS zu aktualisieren

Sie müssen immer sowohl die Zugriffsschlüssel-ID als auch den geheimen Zugriffsschlüssel in der Anmeldeinformation angeben, auch wenn nur ein Wert geändert wird.

Die ersten beiden Befehle legen Variablen fest, die die Anmeldeinformationswerte enthalten. Der Vorgang `UpdateCustomKeyStore` verwendet den Parameter `CustomKeyStoreId`,

um den externen Schlüsselspeicher zu identifizieren. Er verwendet den Parameter `XksProxyAuthenticationCredential` mit seinen Feldern `AccessKeyId` und `RawSecretAccessKey`, um die neue Anmeldeinformation festzulegen. Ersetzen Sie alle Beispielwerte durch tatsächliche Werte für Ihren externen Schlüsselspeicher.

```
$ 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
```

Ändern des Proxy-URI-Pfads

Im folgenden Beispiel wird der Proxy-URI-Pfad (`XksProxyUriPath`) aktualisiert. Die Kombination aus dem Proxy-URI-Endpunkt und dem Proxy-URI-Pfad muss für das AWS-Konto und die Region eindeutig sein. Ersetzen Sie alle Beispielwerte durch tatsächliche Werte für Ihren externen Schlüsselspeicher.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-uri-path /kms/xks/v1
```

Umstellen auf Konnektivität eines VPC-Endpunkt-Service

Im folgenden Beispiel wird die -UpdateCustomKeyStoreOperation verwendet, um den Proxy-Konnektivitätstyp des externen Schlüsselspeichers in zu ändern`VPC_ENDPOINT_SERVICE`. Um diese Änderung vorzunehmen, müssen Sie die erforderlichen Werte für die Konnektivität eines VPC-Endpunkt-Service angeben, einschließlich des Namens des VPC-Endpunktservice (`XksProxyVpcEndpointServiceName`) und eines Proxy-URI-Endpunktwerts (`XksProxyUriEndpoint`), der den privaten DNS-Namen für den VPC-Endpunktservice enthält. Ersetzen Sie alle Beispielwerte durch tatsächliche Werte für Ihren externen Schlüsselspeicher.

```
$ 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
```

Umstellen auf Konnektivität eines öffentlichen Endpunkts

Im folgenden Beispiel wird der Proxy-Konnektivitätstyp des externen Schlüsselspeichers in `PUBLIC_ENDPOINT` geändert. Wenn Sie diese Änderung vornehmen, müssen Sie den Wert des Proxy-URI-Endpunkts (`XksProxyUriEndpoint`) aktualisieren. Ersetzen Sie alle Beispielwerte durch tatsächliche Werte für Ihren externen Schlüsselspeicher.

> **Note**
>
> VPC-Endpunktkonnektivität bietet mehr Sicherheit als öffentliche Endpunktkonnektivität. Bevor Sie auf öffentliche Endpunktkonnektivität umstellen, sollten Sie andere Optionen in Betracht ziehen, z. B. den Proxy für den externen Schlüsselspeicher On-Premises zu platzieren und die VPC nur für die Kommunikation zu nutzen.

```
$ 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
```

Anzeigen eines externen Schlüsselspeichers

Sie können externe Schlüsselspeicher in jedem Konto und jeder Region anzeigen, indem Sie die -AWS KMSKonsole oder die -DescribeCustomKeyStoresOperation verwenden.

Beim Anzeigen eines externen Schlüsselspeichers können Sie folgende Informationen sehen:

- Grundlegende Informationen über den Schlüsselspeicher, einschließlich seines Benutzernamens, seiner ID, seines Schlüsselspeichertyps und seines Erstellungsdatums
- Konfigurationsinformationen für den Proxy des externen Schlüsselspeichers, einschließlich des Konnektivitätstyps, des Proxy-URI-Endpunkts und -pfads sowie der Zugriffsschlüssel-ID Ihrer aktuellen Proxy-Anmeldeinformation.
- Wenn der Proxy des externen Schlüsselspeichers die Konnektivität des VPC-Endpunkt-Service verwendet, zeigt die Konsole den Namen des VPC-Endpunktservice an.
- Den aktuellen Verbindungsstatus

  > **Note**
  >
  > Der Verbindungsstatus Disconnected (Verbindung getrennt) bedeutet, dass der externe Schlüsselspeicher noch nie verbunden war oder dass die Verbindung zum Proxy des

externen Schlüsselspeichers absichtlich getrennt wurde. Wenn Ihre Versuche, einen KMS-Schlüssel in einem verbundenen externen Schlüsselspeicher zu verwenden, jedoch fehlschlagen, kann dies auf ein Problem mit dem externen Schlüsselspeicher oder seinem Proxy hinweisen. Weitere Informationen dazu finden Sie unter Fehler bei der Verbindung mit dem externen Schlüsselspeicher.

- Ein Abschnitt Überwachung mit Diagrammen von Amazon- CloudWatch Metriken, die Ihnen helfen, Probleme mit Ihrem externen Schlüsselspeicher zu erkennen und zu beheben. Hilfe bei der Interpretation der Diagramme, ihrer Verwendung bei der Planung und Fehlerbehebung und der Erstellung von CloudWatch Alarmen auf der Grundlage der Metriken in den Diagrammen finden Sie unter Überwachung eines externen Schlüsselspeichers.

Weitere Informationen finden Sie auch unter:

- Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher
- AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail

Themen

- Eigenschaften des externen Schlüsselspeichers
- Anzeigen eines externen Schlüsselsspeichers (Konsole)
- Anzeigen eines externen Schlüsselsspeichers (API)

Eigenschaften des externen Schlüsselspeichers

Die folgenden Eigenschaften eines externen Schlüsselspeichers sind in der AWS KMS Konsole und der DescribeCustomKeyStores Antwort sichtbar.

Eigenschaften eines benutzerdefinierten Schlüsselspeichers

Die folgenden Werte werden im Abschnitt General configuration (Allgemeine Konfiguration) der Detailseite für jeden benutzerdefinierten Schlüsselspeicher angezeigt. Diese Eigenschaften gelten für alle benutzerdefinierten Schlüsselspeicher, einschließlich AWS CloudHSM-Schlüsselspeicher und externe Schlüsselspeicher.

ID des benutzerdefinierten Schlüsselspeichers

Eine eindeutige ID, die AWS KMS dem benutzerdefinierten Schlüsselspeicher zuweist

Name des benutzerdefinierten Schlüsselspeichers

Ein Anzeigename, den Sie dem benutzerdefinierten Schlüsselspeicher bei dessen Erstellung zuweisen. Sie können diesen Wert jederzeit ändern.

Typ des benutzerdefinierten Schlüsselspeichers

Der Typ des benutzerdefinierten Schlüsselspeichers. Gültige Werte sind AWS CloudHSM (`AWS_CLOUDHSM`) oder „Externer Schlüsselspeicher“ (`EXTERNAL_KEY_STORE`). Der Typ kann nach der Erstellung des benutzerdefinierten Schlüsselspeichers nicht mehr geändert werden.

Erstellungsdatum

Das Datum, an dem der benutzerdefinierte Schlüsselspeicher erstellt wurde. Dieses Datum wird in Ortszeit für die AWS-Region angezeigt.

Verbindungsstatus

Zeigt an, ob der benutzerdefinierte Schlüsselspeicher mit dem Unterstützungsschlüsselspeicher verbunden ist. Der Verbindungsstatus ist nur dann `DISCONNECTED`, wenn der benutzerdefinierte Schlüsselspeicher noch nie mit dem Unterstützungsschlüsselspeicher verbunden war oder die Verbindung absichtlich getrennt wurde. Details hierzu finden Sie unter the section called “Verbindungsstatus”.

Konfigurationseigenschaften des externen Schlüsselspeichers

Die folgenden Werte werden im Abschnitt Konfiguration des externen Schlüsselspeicher-Proxys auf der Detailseite für jeden externen Schlüsselspeicher und im `-XksProxyConfiguration`Element der DescribeCustomKeyStores Antwort angezeigt. Eine ausführliche Beschreibung der einzelnen Felder, einschließlich der Anforderungen an die Eindeutigkeit und Hilfe bei der Bestimmung des richtigen Werts für jedes Feld, finden Sie unter the section called “Erfüllen der Voraussetzungen” im Thema Erstellen eines externen Schlüsselspeichers.

Proxy-Konnektivität

Gibt an, ob der externe Schlüsselspeicher Konnektivität eines öffentlichen Endpunkts oder Konnektivität eines VPC-Endpunkt-Service verwendet.

Proxy-URI-Endpunkt

Der Endpunkt, den AWS KMS für die Verbindung zum Proxy Ihres externen Schlüsselspeichers verwendet.

Proxy-URI-Pfad

Der Pfad vom Proxy-URI-Endpunkt, an den AWS KMS Proxy-API-Anfragen sendet.

Proxy-Anmeldeinformation: Zugriffsschlüssel-ID

Teil der Anmeldeinformation für die Proxy-Authentifizierung, die Sie für den Proxy Ihres externen Schlüsselspeichers einrichten. Die Zugriffsschlüssel-ID identifiziert den geheimen Zugriffsschlüssel in der Anmeldeinformation.

AWS KMS verwendet das SigV4-Signierverfahren und die Anmeldeinformation für die Proxy-Authentifizierung, um seine Anforderungen an den Proxy Ihres externen Schlüsselspeichers zu signieren. Die Anmeldeinformation in der Signatur ermöglicht es dem Proxy des externen Schlüsselspeichers, Anforderungen von AWS KMS in Ihrem Namen zu authentifizieren.

Name des VPC-Endpunkt-Service

Der Name des VPC-Endpunkt-Service von Amazon, der Ihren externen Schlüsselspeicher unterstützt. Dieser Wert wird nur angezeigt, wenn der externe Schlüsselspeicher die Konnektivität eines VPC-Endpunkt-Service nutzt. Sie können den Proxy Ihres externen Schlüsselspeichers in der VPC finden oder den VPC-Endpunkt-Service verwenden, um sicher mit dem Proxy Ihres externen Schlüsselspeichers zu kommunizieren.

Anzeigen eines externen Schlüsselsspeichers (Konsole)

Gehen Sie wie folgt vor, um die externen Schlüsselspeicher in einem bestimmten Konto und einer bestimmten Region anzuzeigen:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Zum Anzeigen von detaillierten Informationen über einen externen Schlüsselspeicher wählen Sie den Namen des Schlüsselspeichers aus.

Anzeigen eines externen Schlüsselsspeichers (API)

Um Ihre externen Schlüsselspeicher anzuzeigen, verwenden Sie die -DescribeCustomKeyStoresOperation. Standardmäßig gibt diese Operation alle benutzerdefinierten Schlüsselspeicher im Konto und in der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die Ausgabe auf einen bestimmten benutzerdefinierten Schlüsselspeicher zu begrenzen.

Bei benutzerdefinierten Schlüsselspeichern besteht die Ausgabe aus der ID, dem Namen und dem Typ des benutzerdefinierten Schlüsselspeichers sowie dem Verbindungsstatus des Schlüsselspeichers. Wenn der Verbindungsstatus `FAILED` ist, enthält die Ausgabe auch einen `ConnectionErrorCode`, der den Grund für den Fehler beschreibt. Hilfe bei der Interpretation des `ConnectionErrorCode` für einen externen Schlüsselspeicher finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher.

Für externe Schlüsselspeicher enthält die Ausgabe auch das `XksProxyConfiguration`-Element. Dieses Element enthält den Konnektivitätstyp, den Proxy-URI-Endpunkt, den Proxy-URI-Pfad und die Zugriffsschlüssel-ID der Anmeldeinformation für die Proxy-Authentifizierung.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Beispielsweise gibt der folgende Befehl alle benutzerdefinierten Schlüsselspeicher im Konto und in der Region zurück. Sie können die Parameter `Marker` und `Limit` verwenden, um durch die benutzerdefinierten Schlüsselspeicher in der Ausgabe zu blättern.

```
$ aws kms describe-custom-key-stores
```

Der folgende Befehl verwendet den Parameter `CustomKeyStoreName`, um nur den externen Beispiel-Schlüsselspeicher mit dem Anzeigenamen `ExampleXksPublic` abzurufen. Dieser Beispiel-Schlüsselspeicher verwendet Konnektivität eines öffentlichen Endpunkts. Er ist mit dem Proxy seines externen Schlüsselspeichers verbunden.

```
$ 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"
      }
    }
  ]
}
```

Mit dem folgenden Befehl wird ein Beispiel für einen externen Schlüsselspeicher mit Konnektivität eines VPC-Endpunkt-Service abgerufen. In diesem Beispiel ist der externe Schlüsselspeicher mit dem Proxy seines externen Schlüsselspeichers verbunden.

```
$ 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"
      }
    }
  ]
}
```

Der `ConnectionState` `Disconnected` bedeutet, dass ein externer Schlüsselspeicher nie verbunden war oder dass er absichtlich vom Proxy seines externen Schlüsselspeichers getrennt wurde. Wenn jedoch der Versuch, einen KMS-Schlüssel in einem verbundenen externen Schlüsselspeicher zu verwenden, fehlschlägt, kann dies auf ein Problem mit dem Proxy des externen Schlüsselspeichers oder anderen externen Komponenten hinweisen.

Wenn der `ConnectionState` des externen Schlüsselspeichers `FAILED` lautet, enthält die `DescribeCustomKeyStores`-Antwort ein `ConnectionErrorCode`-Element, das den Grund für den Fehler angibt.

In der folgenden Ausgabe bedeutet der Wert `XKS_PROXY_TIMED_OUT` beispielsweise, dass AWS KMS eine Verbindung zum Proxy des externen Schlüsselspeichers herstellen kann, die Verbindung jedoch fehlgeschlagen ist, weil der Proxy des externen Schlüsselspeichers AWS KMS nicht in der vorgesehenen Zeit geantwortet hat. Wenn Sie diesen Verbindungsfehlercode wiederholt sehen, benachrichtigen Sie den Proxy-Anbieter Ihres externen Schlüsselspeichers. Weitere Informationen zu diesem und anderen Verbindungsfehlern finden Sie unter Fehlerbehebung bei externen Schlüsselspeichern.

```
$ 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"
      }
    }
  ]
}
```

Überwachung eines externen Schlüsselspeichers

AWS KMS erfasst Metriken für jede Interaktion mit einem externen Schlüsselspeicher und veröffentlicht sie in Ihrem CloudWatch Konto. Diese Metriken werden verwendet, um die Diagramme im Überwachungsabschnitt der Detailseite für jeden externen Schlüsselspeicher zu erstellen. Im folgenden Thema wird beschrieben, wie Sie die Diagramme verwenden, um Betriebs- und Konfigurationsprobleme zu identifizieren und zu beheben, die sich auf Ihren externen Schlüsselspeicher auswirken. Wir empfehlen, die CloudWatch Metriken zu verwenden, um Alarme

einzurichten, die Sie benachrichtigen, wenn Ihr externer Schlüsselspeicher nicht wie erwartet funktioniert. Weitere Informationen finden Sie unter Überwachung mit Amazon CloudWatch.

Themen

- Anzeigen der Diagramme
- Interpretieren der Diagramme
- Festlegen von Alarmen

Anzeigen der Diagramme

Sie können die Diagramme auf verschiedenen Detailebenen anzeigen. Standardmäßig verwendet jedes Diagramm einen Zeitbereich von drei Stunden und einen Aggregationszeitraum von fünf Minuten. Sie können die Diagrammansicht innerhalb der Konsole anpassen, aber Ihre Änderungen werden auf die Standardeinstellungen zurückgesetzt, wenn die Detailseite des externen Schlüsselspeichers geschlossen oder der Browser aktualisiert wird. Hilfe zur Amazon- CloudWatch Terminologie finden Sie unter Amazon- CloudWatch Konzepte.

Anzeigen von Datenpunktdetails

Die Daten in jedem Diagramm werden von AWS KMS-Metriken gesammelt. Um weitere Informationen zu einem bestimmten Datenpunkt anzuzeigen, halten Sie die Maus auf den Datenpunkt im Liniendiagramm. Daraufhin wird ein Popup-Fenster mit weiteren Informationen zu der Metrik angezeigt, von der das Diagramm abgeleitet wurde. Jedes Listenelement zeigt den an diesem Datenpunkt aufgezeichneten Dimensionswert an. Das Pop-up zeigt einen Nullwert (–) an, wenn für den Dimensionswert an diesem Datenpunkt keine Metrikdaten verfügbar sind. Einige Diagramme zeichnen mehrere Dimensionen und Werte für einen einzelnen Datenpunkt auf. Andere Diagramme, wie das Zuverlässigkeitsdiagramm, verwenden die von der Metrik erfassten Daten, um einen eindeutigen Wert zu berechnen. Jedes Listenelement ist einer anderen Farbe des Liniendiagramms zugeordnet.

Ändern des Zeitbereichs

Zum Ändern des Zeitbereichs wählen Sie im Überwachungsbereich oben rechts einen der vordefinierten Zeitbereiche aus. Die vordefinierten Zeitbereiche reichen von 1 Stunde bis 1 Woche (1h, 3h, 12h, 1d, 3d oder 1w). Dadurch wird der Zeitbereich für alle Diagramme angepasst. Wenn Sie ein bestimmtes Diagramm in einem anderen Zeitraum anzeigen möchten oder wenn Sie einen benutzerdefinierten Zeitraum festlegen möchten, vergrößern Sie das Diagramm oder zeigen Sie es in der Amazon- CloudWatch Konsole an.

Vergrößern von Diagrammen

Mit der Mini-Map-Zoomfunktion können Sie sich auf Abschnitte von Liniendiagrammen und gestapelte Teile der Diagramme konzentrieren, ohne zwischen vergrößerter und verkleinerter Ansicht zu wechseln. Sie können die Mini-Map-Zoomfunktion beispielsweise verwenden, um sich auf einen Spitzenwert in einem Diagramm zu konzentrieren, sodass Sie den Spitzenwert mit anderen Diagrammen im Überwachungsabschnitt auf derselben Zeitachse vergleichen können.

1. Wählen Sie den Bereich des Diagramms aus, auf den Sie sich konzentrieren möchten, und lassen Sie die Maustaste los.
2. Um den Zoom zurückzusetzen, wählen Sie das Symbol Zoom zurücksetzen aus, das wie eine Lupe mit einem Minuszeichen (-) darin aussieht.

Vergrößern eines Diagramms

Wenn Sie ein Diagramm vergrößern möchten, wählen Sie darin das Menüsymbol oben rechts aus und wählen Sie dann Enlarge (Vergrößern). Sie können auch das Vergrößerungssymbol auswählen, das neben dem Menüsymbol angezeigt wird, wenn Sie den Mauszeiger über ein Diagramm bewegen.

Durch die Vergrößerung eines Diagramms können Sie die Ansicht eines Diagramms weiter verändern, indem Sie einen anderen Zeitraum, einen benutzerdefinierten Zeitbereich oder ein Aktualisierungsintervall angeben. Diese Änderungen werden auf die Standardeinstellungen zurückgesetzt, wenn Sie die vergrößerte Ansicht schließen.

Ändern des Zeitraums

1. Wählen Sie das Optionsmenü für den Zeitraum aus. In diesem Menü wird standardmäßig folgender Wert angezeigt: 5 Minuten.
2. Wählen Sie einen Zeitraum. Die vordefinierten Zeiträume reichen von 1 Sekunde bis 30 Tage.

   Sie können beispielsweise eine Ein-Minuten-Ansicht auswählen. Dies kann nützlich sein, wenn Sie Fehler beheben. Sie können auch eine weniger detaillierte Ein-Stunden-Ansicht auswählen. Dies kann nützlich sein, wenn Sie einen größeren Zeitraum anzeigen (z. B. 3 Tage), damit Sie Trends über die Zeit anzeigen können. Weitere Informationen finden Sie unter Zeiträume im Amazon- CloudWatch Benutzerhandbuch.

Ändern des Zeitraums oder Zeitzone

1. Wählen Sie einen der vordefinierten Zeitbereiche, die von 1 Stunde bis 1 Woche reichen (1h, 3h, 12h, 1d, 3d oder 1w). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
2. Wählen Sie Custom (Benutzerdefiniert) aus.
   a. Zeitraum: Wählen Sie die Registerkarte Absolute (Absolut) im Feld oben links aus. Verwenden Sie die Kalenderauswahl oder die Textfelder, um einen Zeitraum festzulegen.
   b. Zeitzone: Wählen Sie das Dropdown-Menü im Feld oben rechts aus. Sie können die Zeitzone auf UTC oder Local time zone (lokale Zeitzone) ändern.
3. Wählen Sie nach dem Festlegen eines Zeitraums Apply (Anwenden) aus.

Ändern der Aktualisierungshäufigkeit der Daten in Ihrem Diagramm

1. Wählen Sie oben rechts das Menü Refresh options (Aktualisierungsoptionen) aus.
2. Wählen Sie ein Aktualisierungsintervall (Aus, 10 Sekunden, 1 Minute, 2 Minuten, 5 Minuten oder 15 Minuten).

Diagramme in der Amazon- CloudWatch Konsole anzeigen

Die Diagramme im Überwachungsabschnitt werden von vordefinierten Metriken abgeleitet, die in Amazon AWS KMS veröffentlicht CloudWatch. Sie können sie in der CloudWatch Konsole öffnen und in CloudWatch Dashboards speichern. Wenn Sie mehrere externe Schlüsselspeicher haben, können Sie die entsprechenden Diagramme in öffnen CloudWatch und sie in einem einzigen Dashboard speichern, um ihren Zustand und ihre Nutzung zu vergleichen.

Zum CloudWatch Dashboard hinzufügen

Wählen Sie Zu Dashboard hinzufügen in der oberen rechten Ecke aus, um alle Diagramme zu einem Amazon- CloudWatch Dashboard hinzuzufügen. Sie können entweder ein vorhandenes Dashboard auswählen oder ein neues erstellen. Informationen zur Verwendung dieses Dashboards zum Erstellen benutzerdefinierter Ansichten der Diagramme und Alarme finden Sie unter Verwenden von Amazon CloudWatch-Dashboards im Amazon- CloudWatch Benutzerhandbuch.

Anzeigen in - CloudWatch Metriken

Wählen Sie das Menüsymbol in der oberen rechten Ecke eines einzelnen Diagramms und wählen Sie In Metriken anzeigen, um dieses Diagramm in der Amazon CloudWatch-Konsole anzuzeigen. Von

der CloudWatch Konsole aus können Sie dieses einzelne Diagramm zu einem Dashboard hinzufügen und Zeitbereiche, Zeiträume und Aktualisierungsintervalle ändern. Weitere Informationen finden Sie unter Grafisches Darstellen von Metriken im Amazon- CloudWatch Benutzerhandbuch.

Interpretieren der Diagramme

AWS KMS bietet mehrere Diagramme zur Überwachung des Zustands Ihres externen Schlüsselspeichers innerhalb der AWS KMS-Konsole. Diese Diagramme werden automatisch konfiguriert und von AWS KMS-Metriken abgeleitet.

Die Diagrammdaten werden als Teil der Aufrufe gesammelt, die Sie an Ihren externen Schlüsselspeicher und Ihre externen Schlüssel richten. Es kann sein, dass Sie in einem Zeitraum, in dem Sie keine Aufrufe getätigt haben, Daten in den Diagrammen sehen. Diese Daten stammen von den regelmäßigen `GetHealthStatus`-Aufrufen, die AWS KMS in Ihrem Namen durchführt, um den Status Ihres externen Schlüsselspeicher-Proxys und Ihres externen Schlüsselmanagers zu überprüfen. Wenn Ihre Diagramme die Meldung No data available (Keine Daten verfügbar) anzeigen, wurden in diesem Zeitraum keine Anrufe aufgezeichnet oder Ihr externer Schlüsselspeicher befindet sich im Status `DISCONNECTED`. Möglicherweise können Sie den Zeitpunkt der Trennung Ihres externen Schlüsselspeichers ermitteln, indem Sie Ihre Ansicht auf einen breiteren Zeitbereich einstellen.

Themen

- Anfragen insgesamt
- Zuverlässigkeit
- Latency
- 5 häufigste Ausnahmen
- Gültigkeitsdauer des Zertifikats in Tagen

Anfragen insgesamt

Die Gesamtzahl der AWS KMS-Anforderungen, die für einen bestimmten externen Schlüsselspeicher während eines bestimmten Zeitraums empfangen wurden. Anhand dieses Diagramms können Sie feststellen, ob das Risiko einer Drosselung besteht.

AWS KMS empfiehlt, dass Ihr externer Schlüsselmanager in der Lage ist, bis zu 1 800 Anforderungen für kryptografische Vorgänge pro Sekunde zu verarbeiten. Wenn Sie innerhalb von fünf Minuten 540 000 Aufrufe erreichen, besteht das Risiko einer Drosselung.

Mit der Metrik ExternalKeyStoreThrottle können Sie die Anzahl der Anforderungen für kryptografische Vorgänge für KMS-Schlüssel in Ihrem externen Schlüsselspeicher überwachen, die AWS KMS drosselt.

Wenn Sie sehr häufig `KMSInvalidStateException`-Fehler mit einer Meldung erhalten, in der erklärt wird, dass die Anforderung „aufgrund einer sehr hohen Anforderungsrate“ abgelehnt wurde, könnte dies darauf hinweisen, dass Ihr externer Schlüsselmanager oder externer Schlüsselspeicher-Proxy mit der aktuellen Anforderungsrate nicht Schritt halten kann. Verringen Sie wenn möglich Ihre Anforderungsrate. Sie können auch eine Verringerung des Anforderungskontingentwerts für benutzerdefinierte Schlüsselspeicher anfordern. Eine Verringerung dieses Kontingentwerts könnte die Drosselung erhöhen, deutet aber darauf hin, dass AWS KMS übermäßige Anforderungen schnell zurückweist, bevor sie an Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager gesendet werden. Um eine Reduzierung des Kontingents zu beantragen, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

Das Diagramm für die gesamten Anforderungen wird von der XksProxyErrors-Metrik abgeleitet, die Daten über die erfolgreichen und nicht erfolgreichen Antworten erfasst, die AWS KMS von Ihrem externen Schlüsselspeicher-Proxy erhält. Wenn Sie einen bestimmten Datenpunkt anzeigen, zeigt das Popup-Fenster den Wert der `CustomKeyStoreId`-Dimension neben der Gesamtzahl der AWS KMS-Anforderungen an, die an diesem Datenpunkt aufgezeichnet wurden. Die `CustomKeyStoreId` bleibt immer gleich.

Zuverlässigkeit

Der Prozentsatz der AWS KMS-Anforderungen, für die der externe Schlüsselspeicher-Proxy entweder eine erfolgreiche Antwort oder einen nicht wiederholbaren Fehler zurückgegeben hat. Bewerten Sie anhand dieses Diagramms den Betriebszustand Ihres externen Schlüsselspeicher-Proxys.

Wenn das Diagramm einen Wert von weniger als 100 % anzeigt, weist dies auf Fälle hin, in denen der Proxy entweder nicht oder mit einem wiederholbaren Fehler geantwortet hat. Dies kann auf Probleme mit dem Netzwerk, Langsamkeit des externen Schlüsselspeicher-Proxys oder des externen Schlüsselmanagers oder auf Implementierungsfehler hindeuten.

Wenn die Anforderung eine falsche Anmeldeinformation enthält und Ihr Proxy mit einer `AuthenticationFailedException` antwortet, zeigt das Diagramm trotzdem 100 % Zuverlässigkeit an, weil der Proxy einen falschen Wert in der API-Anforderung des externen Schlüsselspeicher-Proxys identifiziert hat und der Fehler daher erwartet wurde. Wenn der Prozentsatz Ihres Zuverlässigkeitsdiagramms 100 % beträgt, reagiert Ihr externer Schlüsselspeicher-

Proxy wie erwartet. Wenn das Diagramm einen Wert von weniger als 100 % anzeigt, hat der Proxy entweder mit einem wiederholbaren Fehler geantwortet oder eine Zeitüberschreitung verursacht. Wenn der Proxy beispielsweise aufgrund einer sehr hohen Anzahl von Anforderungen mit einer `ThrottlingException` antwortet, wird ein niedrigerer Zuverlässigkeitsgrad angezeigt, da der Proxy nicht in der Lage war, ein spezifisches Problem in der Anforderung zu identifizieren, das zum Fehlschlagen führte. Dies liegt daran, dass es sich bei wiederholbaren Fehlern wahrscheinlich um vorübergehende Probleme handelt, die durch einen erneuten Versuch der Anforderung behoben werden können.

Die folgenden Fehlerantworten senken den Zuverlässigkeitsgrad. Mit dem Diagramm 5 häufigste Ausnahmen und der Metrik XksProxyErrors können Sie weiter überwachen, wie häufig Ihr Proxy jeden wiederholbaren Fehler zurückgibt.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

Das Zuverlässigkeitsdiagramm wird von der XksProxyErrors-Metrik abgeleitet, die Daten über die erfolgreichen und nicht erfolgreichen Antworten erfasst, die AWS KMS von Ihrem externen Schlüsselspeicher-Proxy erhält. Der Zuverlässigkeitsprozentsatz sinkt nur, wenn die Antwort den `ErrorType`-Wert `Retryable` hat. Wenn Sie einen bestimmten Datenpunkt anzeigen, zeigt das Popup-Fenster den Wert der `CustomKeyStoreId`-Dimension neben dem Prozentsatz der Zuverlässigkeit für AWS KMS-Anforderungen an, die an diesem Datenpunkt aufgezeichnet wurden. Die `CustomKeyStoreId` bleibt immer gleich.

Wir empfehlen, die -XksProxyErrorsMetrik zu verwenden, um einen CloudWatch Alarm zu erstellen, der Sie über potenzielle Netzwerkprobleme benachrichtigt, indem er Sie warnt, wenn mehr als fünf wiederholbare Fehler in einem Zeitraum von einer Minute aufgezeichnet werden. Weitere Informationen finden Sie unter Erstellen eines Amazon- CloudWatch Alarms für wiederholbare Fehler.

Latency

Die Anzahl der Millisekunden, die ein externer Schlüsselspeicher-Proxy für die Antwort auf eine AWS KMS-Anforderung benötigt. Verwenden Sie dieses Diagramm, um die Leistung Ihres externen Schlüsselspeicher-Proxys und externen Schlüsselmanagers zu bewerten.

AWS KMS erwartet, dass der externe Schlüsselspeicher-Proxy auf jede Anforderung innerhalb von 250 Millisekunden antwortet. Im Falle von Netzwerk-Timeouts wiederholt AWS KMS die Anforderung einmal. Wenn der Proxy ein zweites Mal fehlschlägt, entspricht die aufgezeichnete Latenz der kombinierten Timeout-Grenze für beide Anforderungen und das Diagramm zeigt etwa 500 Millisekunden an. In allen anderen Fällen, in denen der Proxy nicht innerhalb der Timeout-Grenze von 250 Millisekunden antwortet, beträgt die aufgezeichnete Latenzzeit 250 Millisekunden. Wenn der Proxy bei Verschlüsselungs- und Entschlüsselungsvorgängen häufig eine Zeitüberschreitung aufweist, wenden Sie sich an Ihren externen Proxy-Administrator. Hilfe zur Behebung von Latenzproblemen finden Sie unter Latenz- und Zeitüberschreitungsfehler.

Langsame Antworten können auch darauf hinweisen, dass Ihr externer Schlüsselmanager den aktuellen Datenverkehr nicht bewältigen kann. AWS KMS empfiehlt, dass Ihr externer Schlüsselmanager in der Lage ist, bis zu 1 800 Anforderungen für kryptografische Vorgänge pro Sekunde zu verarbeiten. Wenn Ihr externer Schlüsselmanager die Rate von 1 800 Anforderungen pro Sekunde nicht bewältigen kann, sollten Sie eine Verringerung Ihrer Anforderungsquote für KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher anfordern. Anforderungen für kryptografische Vorgänge, die die KMS-Schlüssel in Ihrem externen Schlüsselspeicher verwenden, schlagen schnell mit einer Drosselungsausnahme fehl, anstatt verarbeitet und später von Ihrem externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager abgelehnt zu werden.

Das Latenzdiagramm wird von der XksProxyLatency-Metrik abgeleitet. Wenn Sie einen bestimmten Datenpunkt anzeigen, zeigt das Popup-Fenster die entsprechenden Werte der Dimensionen `KmsOperation` und `XksOperation` sowie die durchschnittliche Latenz an, die für die Vorgänge an diesem Datenpunkt aufgezeichnet wurde. Die Listenelemente sind von der höchsten zur niedrigsten Latenz sortiert.

Wir empfehlen, die -XksProxyLatencyMetrik zu verwenden, um einen CloudWatch Alarm zu erstellen, der Sie benachrichtigt, wenn sich Ihre Latenz dem Timeout-Limit nähert. Weitere Informationen finden Sie unter Erstellen eines Amazon- CloudWatch Alarms für ein Reaktions-Timeout.

5 häufigste Ausnahmen

Die fünf häufigsten Ausnahmen für fehlgeschlagene kryptografische und verwaltungstechnische Vorgänge innerhalb eines bestimmten Zeitraums. Verwenden Sie dieses Diagramm, um die häufigsten Fehler zu verfolgen, sodass Sie Ihre Entwicklungsarbeit nach Prioritäten ordnen können.

Diese Zahl umfasst Ausnahmen, die AWS KMS vom externen Schlüsselspeicher-Proxy erhalten hat, und die `XksProxyUnreachableException`, die AWS KMS intern zurückgibt, wenn keine Kommunikation mit dem externen Schlüsselspeicher-Proxy hergestellt werden kann.

Hohe Raten wiederholbarer Fehler können auf Netzwerkfehler hinweisen, während hohe Raten nicht wiederholbarer Fehler auf ein Problem mit der Konfiguration Ihres externen Schlüsselspeichers hinweisen können. Eine hohe Anzahl von `AuthenticationFailedExceptions` weist beispielsweise auf eine Diskrepanz zwischen den in AWS KMS konfigurierten Authentifizierungsanmeldeinformationen und dem Proxy des externen Schlüsselspeichers hin. Informationen zur Konfiguration Ihres externen Schlüsselspeichers finden Sie unter Anzeigen eines externen Schlüsselspeichers. Informationen zum Bearbeiten der Einstellungen Ihres externen Schlüsselspeichers finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.

Die Ausnahmen, die AWS KMS vom externen Schlüsselspeicher-Proxy empfängt, unterscheiden sich von den Ausnahmen, die von AWS KMS zurückgegeben werden, wenn ein Vorgang fehlschlägt. Kryptografische AWS KMS-Operationen geben eine `KMSInvalidStateException` für alle Fehler im Zusammenhang mit der externen Konfiguration oder dem Verbindungsstatus des externen Schlüsselspeichers zurück. Identifizieren Sie das Problem anhand des zugehörigen Fehlermeldungstexts.

Die folgende Tabelle enthält die Ausnahmen, die im Diagramm der 5 häufigsten Ausnahmen erscheinen können, und die entsprechenden Ausnahmen, die AWS KMS an Sie zurückgibt.

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
Nicht wiederholbar	**`AccessDeniedException`** Hilfe zur Problembehebung finden Sie unter Probleme mit der Proxy-Autorisierung.	**`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`AuthenticationFailedException`** Hilfe zur Problembehebung finden Sie unter Fehler mit der	**`XksProxyIncorrectAuthenticationCredentialException`** als Antwort auf `CreateCustomKeyStore` - und

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
	Anmeldeinformation für die Authentifizierung.	`UpdateCustomKeyStore` -Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Wiederholbar	**`DependencyTimeoutException`** Hilfe zur Problembehebung finden Sie unter Latenz- und Zeitüberschreitungsfehler.	**`XksProxyUriUnreachableException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` -Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
Wiederholbar	**`InternalException`** Der Proxy des externen Schlüsselspeichers hat die Anforderung abgelehnt, weil er nicht mit dem externen Schlüsselmanager kommunizieren kann. Stellen Sie sicher, dass die Proxykonfiguration für den externen Schlüsselspeicher korrekt ist und dass der externe Schlüsselmanager verfügbar ist.	**`XksProxyInvalidResponseException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` -Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`InvalidCiphertextException`** Hilfe zur Problembehebung finden Sie unter Entschlüsselungsfehler.	**`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`InvalidKeyUsageException`** Hilfe zur Problembehebung finden Sie unter Kryptografische Operationsfehler für den externen Schlüssel.	**`XksKeyInvalidConfigurationException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
Nicht wiederholbar	**`InvalidStateException`** Hilfe zur Problembehebung finden Sie unter Kryptografische Operationsfehler für den externen Schlüssel.	**`XksKeyInvalidConfigurationException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`InvalidUriPathException`** Hilfe zur Problembehebung finden Sie unter Allgemeine Konfigurationsfehler.	**`XksProxyInvalidConfigurationException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` -Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`KeyNotFoundException`** Hilfe zur Problembehebung finden Sie unter Fehler mit externen Schlüsseln.	**`XksKeyNotFoundException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
Wiederholbar	**`ThrottlingException`** Der Proxy des externen Schlüsselspeichers hat die Anforderung aufgrund einer sehr hohen Anzahl von Anforderungen abgelehnt. Verringern Sie die Häufigkeit Ihrer Aufrufe mit KMS-Schlüsseln in diesem externen Schlüsselspeicher.	**`XksProxyUriUnreachableException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` -Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Nicht wiederholbar	**`UnsupportedOperationException`** Hilfe zur Problembehebung finden Sie unter Kryptografische Operationsfehler für den externen Schlüssel.	**`XksKeyInvalidResponseException`** als Antwort auf `CreateKey` -Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.

Fehlertyp	Im Diagramm angezeigte Ausnahme	Ausnahme, die AWS KMS an Sie zurückgibt
Nicht wiederholbar	**`ValidationException`** Hilfe zur Problembehebung finden Sie unter Proxy-Probleme.	**`XksProxyInvalidResponseException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` - Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` - Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.
Wiederholbar	**`XksProxyUnreachableException`** Wenn dieser Fehler wiederholt auftritt, überprüfen Sie, ob Ihr externer Schlüsselspeicher-Proxy aktiv und mit dem Netzwerk verbunden ist und ob sein URI-Pfad und Endpunkt-URI oder der Name des VPC-Service in Ihrem externen Schlüsselspeicher korrekt sind.	**`XksProxyUriUnreachableException`** als Antwort auf `CreateCustomKeyStore` - und `UpdateCustomKeyStore` - Vorgänge. **`CustomKeyStoreInvalidStateException`** als Antwort auf `CreateKey` - Vorgänge. **`KMSInvalidStateException`** als Antwort auf kryptografische Vorgänge.

Das Diagramm mit den fünf wichtigsten Ausnahmen wird von der XksProxyErrors-Metrik abgeleitet. Wenn Sie einen bestimmten Datenpunkt anzeigen, zeigt das Popup-Fenster den Wert der Dimension

`ExceptionName` zusammen mit der Anzahl, wie oft die Ausnahme an diesem Datenpunkt aufgezeichnet wurde. Die fünf Elemente der Liste sind in der Reihenfolge der häufigsten Ausnahme bis zur geringsten geordnet.

Wir empfehlen, die -XksProxyErrorsMetrik zu verwenden, um einen CloudWatch Alarm zu erstellen, der Sie über potenzielle Konfigurationsprobleme benachrichtigt, indem er Sie benachrichtigt, wenn mehr als fünf nicht wiederholbare Fehler in einem Zeitraum von einer Minute aufgezeichnet werden. Weitere Informationen finden Sie unter Erstellen eines Amazon- CloudWatch Alarms für nicht wiederholbare Fehler.

Gültigkeitsdauer des Zertifikats in Tagen

Die Anzahl der Tage, bis das TLS-Zertifikat für den Proxy-Endpunkt Ihres externen Schlüsselspeichers (`XksProxyUriEndpoint`) abläuft. Verwenden Sie dieses Diagramm zur Überwachung des bevorstehenden Ablaufs Ihres TLS-Zertifikats.

Wenn das Zertifikat abläuft, kann AWS KMS nicht mit dem Proxy des externen Schlüsselspeichers kommunizieren. Der Zugriff auf alle durch KMS-Schlüssel geschützten Daten in Ihrem externen Schlüsselspeicher ist dann erst wieder möglich, wenn Sie das Zertifikat erneuern.

Das Diagramm mit der Gültigkeitsdauer des Zertifikats in Tagen wird von der XksProxyCertificateDaysToExpire-Metrik abgeleitet. Wir empfehlen dringend, diese Metrik zu verwenden, um einen CloudWatch Alarm zu erstellen, der Sie über den bevorstehenden Ablauf benachrichtigt. Der Ablauf des Zertifikats kann den Zugriff auf Ihre verschlüsselten Ressourcen verhindern. Stellen Sie den Alarm so ein, dass Ihre Organisation Zeit hat, das Zertifikat zu erneuern, bevor es abläuft. Weitere Informationen finden Sie unter Erstellen eines Amazon- CloudWatch Alarms für den Ablauf von Zertifikaten.

Festlegen von Alarmen

Die Diagramme im Überwachungsabschnitt geben einen Überblick über den Zustand Ihrer externen Schlüsselspeicher und KMS-Schlüssel in externen Schlüsselspeichern für einen bestimmten Zeitraum. Sie können jedoch Amazon- CloudWatch Alarme basierend auf Metriken für externe Schlüsselspeicher erstellen, um Sie zu benachrichtigen, wenn ein Metrikwert einen von Ihnen angegebenen Schwellenwert überschreitet. Der Alarm kann die E-Mail-Nachricht an ein Amazon Simple Notification Service (Amazon SNS)-Thema oder eine Richtlinie von Amazon EC2 Auto Scaling senden. Ausführliche Informationen zu CloudWatch Alarmen finden Sie unter Verwenden von Amazon- CloudWatch Alarmen im Amazon- CloudWatch Benutzerhandbuch.

Bevor Sie einen Amazon- CloudWatch Alarm erstellen, benötigen Sie ein Amazon SNS-Thema. Weitere Informationen finden Sie unter Erstellen eines Amazon SNS-Themas im Amazon-CloudWatch Benutzerhandbuch.

Themen

- Erstellen eines Amazon- CloudWatch Alarms für den Ablauf von Zertifikaten
- Erstellen eines Amazon- CloudWatch Alarms für ein Reaktions-Timeout
- Erstellen eines Amazon- CloudWatch Alarms für wiederholbare Fehler
- Erstellen eines Amazon- CloudWatch Alarms für nicht wiederholbare Fehler

Erstellen eines Amazon- CloudWatch Alarms für den Ablauf von Zertifikaten

Dieser Alarm verwendet die XksProxyCertificateDaysToExpire Metrik, die in AWS KMS veröffentlicht, CloudWatch um den voraussichtlichen Ablauf des TLS-Zertifikats aufzuzeichnen, das Ihrem externen Schlüsselspeicher-Proxy-Endpunkt zugeordnet ist. Sie können nicht einen einzigen Alarm für alle externen Schlüsselspeicher in Ihrem Konto oder einen Alarm für externe Schlüsselspeicher erstellen, die Sie möglichweise in Zukunft erstellen.

Wir empfehlen, den Alarm so einzustellen, dass Sie 10 Tage vor Ablauf Ihres Zertifikats gewarnt werden, aber Sie sollten den Schwellenwert festlegen, der Ihren Bedürfnissen am besten entspricht.

Den Alarm erstellen

Folgen Sie den Anweisungen unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert mit den folgenden erforderlichen Werten. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrik auswählen	Wählen Sie KMS und dann XKS Proxy Certificate Metrics (XKS-Proxy-Zertifikatsmetriken). Aktivieren Sie das Kontrollkästchen neben dem `XksProxyCertificateName` , den Sie überwachen möchten. Wählen Sie dann Select Metric (Metrik auswählen) aus.
Statistik	Minimum

Feld	Value (Wert)
Intervall	5 Minuten
Threshold-Typ	Statisch
Immer, wenn ...	Immer wenn `Lower` als XksProxyCertificateDaysToExpire ist`10`.

Erstellen eines Amazon- CloudWatch Alarms für ein Reaktions-Timeout

Dieser Alarm verwendet die XksProxyLatency Metrik, die in AWS KMS CloudWatch veröffentlicht, um die Anzahl der Millisekunden aufzuzeichnen, die ein externer Schlüsselspeicher-Proxy benötigt, um auf eine -AWS KMSAnforderung zu antworten. Sie können nicht einen einzigen Alarm für alle externen Schlüsselspeicher in Ihrem Konto oder einen Alarm für externe Schlüsselspeicher erstellen, die Sie möglichweise in Zukunft erstellen.

AWS KMS erwartet, dass der externe Schlüsselspeicher-Proxy auf jede Anforderung innerhalb von 250 Millisekunden antwortet. Wir empfehlen, einen Alarm einzustellen, der Sie benachrichtigt, wenn Ihr externer Schlüsselspeicher-Proxy länger als 200 Millisekunden braucht, um zu antworten, aber Sie sollten den Schwellenwert festlegen, der Ihren Bedürfnissen am besten entspricht.

Den Alarm erstellen

Folgen Sie den Anweisungen unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert mit den folgenden erforderlichen Werten. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrik auswählen	Wählen Sie KMS und dann XKS Proxy Latency Metrics (XKS-Proxy-Latenzm etriken). Aktivieren Sie das Kontrollkästchen neben dem `KmsOperation` , den Sie überwachen möchten. Wählen Sie dann Select Metric (Metrik auswählen) aus.
Statistik	Durchschnitt

Feld	Value (Wert)
Intervall	5 Minuten
Threshold-Typ	Statisch
Immer, wenn ...	Immer wenn `Greater` als XksProxyLatency ist`200`.

Erstellen eines Amazon- CloudWatch Alarms für wiederholbare Fehler

Dieser Alarm verwendet die XksProxyErrors Metrik, die in AWS KMS veröffentlicht, CloudWatch um die Anzahl der Ausnahmen im Zusammenhang mit AWS KMS Anfragen an Ihren externen Schlüsselspeicher-Proxy aufzuzeichnen. Sie können nicht einen einzigen Alarm für alle externen Schlüsselspeicher in Ihrem Konto oder einen Alarm für externe Schlüsselspeicher erstellen, die Sie möglichweise in Zukunft erstellen.

Wiederholbare Fehler verringern Ihren Zuverlässigkeitsgrad und können auf Netzwerkfehler hinweisen. Wir empfehlen, einen Alarm einzustellen, der Sie warnt, wenn mehr als fünf wiederholbare Fehler innerhalb einer Minute aufgezeichnet werden, aber Sie sollten den Schwellenwert festlegen, der Ihren Bedürfnissen am besten entspricht.

Folgen Sie den Anweisungen unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert mit den folgenden erforderlichen Werten. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrik auswählen	Wählen Sie die Registerkarte Queries (Abfragen) aus. Wählen Sie `AWS/KMS` als Namespace aus. Geben Sie `SUM(XksProxyErrors)` als Metriknamen ein. Geben Sie `ErrorType = Retryable` bei Filter by (Filtern nach) ein. Wählen Sie Ausführen aus. Wählen Sie dann Select Metric (Metrik auswählen) aus.

Feld	Value (Wert)
Label (Bezeichnung)	*Wiederholbare Fehler*
Intervall	1 Minute
Threshold-Typ	Statisch
Immer, wenn ...	Immer wenn q1 `Greater` als 5 ist.

Erstellen eines Amazon- CloudWatch Alarms für nicht wiederholbare Fehler

Dieser Alarm verwendet die XksProxyErrors Metrik, die in AWS KMS veröffentlicht, CloudWatch um die Anzahl der Ausnahmen im Zusammenhang mit AWS KMS Anfragen an Ihren externen Schlüsselspeicher-Proxy aufzuzeichnen. Sie können nicht einen einzigen Alarm für alle externen Schlüsselspeicher in Ihrem Konto oder einen Alarm für externe Schlüsselspeicher erstellen, die Sie möglichweise in Zukunft erstellen.

Nicht wiederholbare Fehler können auf ein Problem mit der Konfiguration Ihres externen Schlüsselspeichers hinweisen. Wir empfehlen, einen Alarm einzustellen, der Sie warnt, wenn mehr als fünf nicht wiederholbare Fehler innerhalb einer Minute aufgezeichnet werden, aber Sie sollten den Schwellenwert festlegen, der Ihren Bedürfnissen am besten entspricht.

Folgen Sie den Anweisungen unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert mit den folgenden erforderlichen Werten. Übernehmen Sie für andere Felder die Standardwerte und geben Sie Namen wie gewünscht an.

Feld	Value (Wert)
Metrik auswählen	Wählen Sie die Registerkarte Queries (Abfragen) aus. Wählen Sie `AWS/KMS` als Namespace aus. Geben Sie `SUM(XksProxyErrors)` als Metriknamen ein. Geben Sie `ErrorType = Non-retryable` bei Filter by (Filtern nach) ein.

Feld	Value (Wert)
	Wählen Sie Ausführen aus. Wählen Sie dann Select Metric (Metrik auswählen) aus.
Label (Bezeichnung)	*Nicht wiederholbare Fehler*
Intervall	1 Minute
Threshold-Typ	Statisch
Immer, wenn ...	Immer wenn q1 `Greater` als 5 ist.

Herstellen und Trennen der Verbindung eines externen Schlüsselspeichers

Neue externe Schlüsselspeicher sind nicht verbunden. Um AWS KMS keys in Ihrem externen Schlüsselspeicher zu erstellen und zu verwenden, müssen Sie Ihren externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbinden. Sie können Ihren externen Schlüsselspeicher jederzeit verbinden und trennen und seinen Verbindungsstatus anzeigen.

Wenn Ihr externer Schlüsselspeicher nicht verbunden ist, kann AWS KMS nicht mit dem Proxy Ihres externen Schlüsselspeichers kommunizieren. Daher können Sie Ihren externen Schlüsselspeicher und seine vorhandenen KMS-Schlüssel anzeigen und verwalten. Sie können jedoch keine KMS-Schlüssel in Ihrem externen Schlüsselspeicher erstellen oder dessen KMS-Schlüssel in kryptografischen Vorgängen verwenden. Es kann sein, dass Sie die Verbindung Ihres externen Schlüsselspeichers irgendwann trennen müssen, z. B. wenn Sie seine Eigenschaften bearbeiten. Durch das Trennen des Schlüsselspeichers kann der Betrieb von AWS-Services unterbrochen werden, die seine KMS-Schlüssel verwenden.

Sie müssen Ihren externen Schlüsselspeicher nicht verbinden. Sie können die Verbindung eines externen Schlüsselspeichers auf unbestimmte Zeit getrennt lassen und die Verbindung nur herstellen, wenn Sie den Schlüsselspeicher verwenden müssen. Möglicherweise möchten Sie die Verbindung jedoch von Zeit zu Zeit testen, um zu prüfen, ob die Einstellungen korrekt sind und eine Verbindungsherstellung möglich ist.

Wenn Sie die Verbindung zu einem benutzerdefinierten Schlüsselspeicher trennen, werden die KMS-Schlüssel im Schlüsselspeicher sofort unbrauchbar (je nach letztendlicher Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel geschützten Datenschlüsseln verschlüsselt wurden,

sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, von denen viele Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

> **Note**
>
> Externe Schlüsselspeicher haben nur dann den Status `DISCONNECTED`, wenn der Schlüsselspeicher noch nie verbunden wurde oder Sie ihn explizit trennen. Der Status `CONNECTED` bedeutet nicht, dass der externe Schlüsselspeicher oder seine unterstützenden Komponenten effizient funktionieren. Informationen über die Leistung der Komponenten Ihres externen Schlüsselspeichers finden Sie in den Diagrammen im Abschnitt Monitoring (Überwachung) auf der Detailseite für jeden externen Schlüsselspeicher. Details hierzu finden Sie unter Überwachung eines externen Schlüsselspeichers.
> Ihr externer Schlüsselmanager bietet möglicherweise zusätzliche Methoden zum Stoppen und Neustarten der Kommunikation zwischen Ihrem externen AWS KMS-Schlüsselspeicher und Ihrem externen Schlüsselspeicher-Proxy oder zwischen Ihrem externen Schlüsselspeicher-Proxy und dem externen Schlüsselmanager. Weitere Informationen finden Sie in der Dokumentation zum externen Schlüsselmanager.

Themen

Verbinden eines externen Schlüsselspeichers

Wenn Ihr externer Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbunden ist, können Sie KMS-Schlüssel in Ihrem externen Schlüsselspeicher erstellen und seine vorhandenen KMS-Schlüssel in kryptografischen Vorgängen verwenden.

Der Prozess, der einen externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbindet, unterscheidet sich je nach der Konnektivität des externen Schlüsselspeichers.

- Wenn Sie einen externen Schlüsselspeicher mit der Konnektivität eines öffentlichen Endpunkts verbinden, AWS KMS sendet eine GetHealthStatus Anforderung an den externen Schlüsselspeicher-Proxy, um den Proxy-URI-Endpunkt, den Proxy-URI-Pfad und die Proxy-Authentifizierungsanmeldeinformationen zu validieren. Eine erfolgreiche Antwort vom Proxy bestätigt, dass der Proxy-URI-Endpunkt und der Proxy-URI-Pfad korrekt und zugänglich sind und dass der Proxy die mit der Proxy-Authentifizierungsanmeldeinformation für den externen Schlüsselspeicher signierte Anforderung authentifiziert hat.
- Wenn Sie einen externen Schlüsselspeicher mit Konnektivität eines VPC-Endpunkt-Service mit seinem externen Schlüsselspeicher-Proxy verbinden, führt AWS KMS die folgenden Schritte aus:
  - Bestätigt, dass die Domain für den im Proxy-URI-Endpunkt angegebenen privaten DNS-Namen verifiziert ist.
  - Erstellt einen Schnittstellen-Endpunkt von einer AWS KMS-VPC zu Ihrem VPC-Endpunkt-Service.
  - Erstellt eine private gehostete Zone für den im Proxy-URI-Endpunkt angegebenen privaten DNS-Namen.
  - Sendet eine GetHealthStatus Anforderung an den externen Schlüsselspeicher-Proxy. Eine erfolgreiche Antwort vom Proxy bestätigt, dass der Proxy-URI-Endpunkt und der Proxy-URI-Pfad korrekt und zugänglich sind und dass der Proxy die mit der Proxy-Authentifizierungsanmeldeinformation für den externen Schlüsselspeicher signierte Anforderung authentifiziert hat.

Der Verbindungsvorgang beginnt mit der Verbindung Ihres benutzerdefinierten Schlüsselspeichers, die Verbindung eines externen Schlüsselspeichers mit seinem externen Proxy dauert jedoch etwa fünf Minuten. Eine Erfolgsmeldung des Verbindungsvorgangs bedeutet nicht, dass der externe Schlüsselspeicher verbunden ist. Um zu bestätigen, dass die Verbindung erfolgreich war, verwenden Sie die -AWS KMSKonsole oder die -DescribeCustomKeyStoresOperation, um den Verbindungsstatus Ihres externen Schlüsselspeichers anzuzeigen.

Wenn der Verbindungsstatus `FAILED` ist, wird in der AWS KMS-Konsole ein Verbindungsfehlercode angezeigt und der `DescribeCustomKeyStore`-Antwort hinzugefügt. Hilfe zur Interpretation von Verbindungsfehlercodes finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher.

Trennen eines externen Schlüsselspeichers

Wenn Sie einen externen Schlüsselspeicher mit Konnektivität eines VPC-Endpunkt-Service von seinem externen Schlüsselspeicher-Proxy trennen, löscht AWS KMS seinen Schnittstellen-Endpunkt zum VPC-Endpunkt-Service und entfernt die Netzwerkinfrastruktur, die zur Unterstützung der Verbindung erstellt wurde. Für externe Schlüsselspeicher mit öffentlicher Endpunktkonnektivität ist kein entsprechender Prozess erforderlich. Diese Aktion wirkt sich weder auf den VPC-Endpunkt-Service oder eine seiner unterstützenden Komponenten noch auf den externen Schlüsselspeicher-Proxy oder externe Komponenten aus.

Während der externe Schlüsselspeicher getrennt ist, sendet AWS KMS keine Anforderungen an den Proxy des externen Schlüsselspeichers. Der Verbindungsstatus des externen Schlüsselspeichers ist `DISCONNECTED`. Die KMS-Schlüssel im getrennten externen Schlüsselspeicher befinden sich in einem `UNAVAILABLE`-Schlüsselzustand (es sei denn, sie sind zum Löschen vorgesehen), was bedeutet, dass sie nicht für kryptografische Vorgänge verwendet werden können. Sie können Ihren externen Schlüsselspeicher und die vorhandenen KMS-Schlüssel jedoch weiterhin anzeigen und verwalten.

Der getrennte Zustand ist als vorübergehend und umkehrbar konzipiert. Sie können die Verbindung Ihres externen Schlüsselspeichers jederzeit wieder herstellen. Normalerweise ist keine Neukonfiguration erforderlich. Wenn sich jedoch Eigenschaften des zugehörigen externen Schlüsselspeicher-Proxys geändert haben, während die Verbindung getrennt war, z. B. die Rotation der Anmeldeinformation für die Proxy-Authentifizierung, müssen Sie die Einstellungen des externen Schlüsselspeichers vor der erneuten Verbindung bearbeiten.

> **Note**
>
> Sämtliche Versuche, KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen oder vorhandene KMS-Schlüssel in kryptografischen Produktionen zu nutzen, schlagen fehl, während der benutzerdefinierte Schlüsselspeicher getrennt ist. Diese Aktion kann verhindern, dass Benutzer vertrauliche Daten speichern und darauf zugreifen.

Um die Auswirkung einer Trennung der Verbindung Ihres externen Schlüsselspeichers besser beurteilen zu können, ermitteln Sie die KMS-Schlüssel im externen Schlüsselspeicher und ihre bisherige Verwendung.

Die Verbindung eines externen Schlüsselspeichers könnte beispielsweise aus folgenden Gründen getrennt werden:

- Zum Bearbeiten seiner Eigenschaften. Sie können den Namen des benutzerdefinierten Schlüsselspeichers, den Proxy-URI-Pfad und die Anmeldeinformation für die Proxy-Authentifizierung bearbeiten, während der externe Schlüsselspeicher verbunden ist. Um jedoch den Typ der Proxy-Konnektivität, den Proxy-URI-Endpunkt oder den Namen des VPC-Endpunkt-Services zu bearbeiten, müssen Sie zunächst die Verbindung des externen Schlüsselspeichers trennen. Details hierzu finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.
- Zum Beenden der gesamten Kommunikation zwischen AWS KMS und dem externen Schlüsselspeicher-Proxy. Sie können die Kommunikation zwischen AWS KMS und Ihrem Proxy auch beenden, indem Sie Ihren Endpunkt oder VPC-Endpunkt-Service deaktivieren. Darüber hinaus bietet Ihr externer Schlüsselspeicher-Proxy oder Ihre Software zur Schlüsselverwaltungs möglicherweise zusätzliche Mechanismen, um zu verhindern, dass AWS KMS mit dem Proxy kommuniziert oder dass der Proxy auf Ihren externen Schlüsselmanager zugreift.
- Zum Deaktivieren aller KMS-Schlüssel im externen Schlüsselspeicher. Sie können KMS-Schlüssel in einem externen Schlüsselspeicher mithilfe der Konsole oder der -Operation deaktivieren und wieder aktivieren. AWS KMS DisableKey Diese Vorgänge werden schnell abgeschlossen (vorbehaltlich einer letztendlichen Konsistenz), beziehen sich jedoch immer nur auf jeweils einen KMS-Schlüssel. Wenn die Verbindung zum externen Schlüsselspeicher unterbrochen wird, ändert sich der Schlüsselstatus aller KMS-Schlüssel im externen Schlüsselspeicher in `Unavailable`, sodass sie in keinem kryptografischen Vorgang verwendet werden können.
- Zur Behebung eines fehlgeschlagenen Verbindungsversuchs. Wenn ein Versuch, eine Verbindung für einen externen Schlüsselspeicher herzustellen, fehlschlägt (Verbindungsstatus des benutzerdefinierten Schlüsselspeichers ist `FAILED`), müssen Sie die Verbindung des externen Schlüsselspeichers trennen, bevor Sie erneut versuchen, eine Verbindung herzustellen.

Verbindungsstatus

Beim Verbinden und Trennen der Verbindung wird der Verbindungsstatus Ihres benutzerdefinierten Schlüsselspeichers geändert. Die Verbindungsstatuswerte sind für AWS CloudHSM-Schlüsselspeicher und externe Schlüsselspeicher identisch.

Um den Verbindungsstatus Ihres benutzerdefinierten Schlüsselspeichers anzuzeigen, verwenden Sie die -DescribeCustomKeyStoresOperation oder die -AWS KMSKonsole. Der Verbindungsstatus wird in jeder Tabelle des benutzerdefinierten Schlüsselspeichers, im Abschnitt General configuration (Allgemeine Konfiguration) der Detailseite jedes benutzerdefinierten Schlüsselspeichers und auf der Registerkarte Cryptographic configuration (Kryptografische Konfiguration) der KMS-Schlüssel in

einem benutzerdefinierten Schlüsselspeicher angezeigt. Details dazu finden Sie unter Anzeigen eines AWS CloudHSM-Schlüsselspeichers und Anzeigen eines externen Schlüsselspeichers.

Ein benutzerdefinierter Schlüsselspeicher kann einen der folgenden Verbindungsstatus haben:

- `CONNECTED`: Der benutzerdefinierte Schlüsselspeicher ist mit seinem Unterstützungsschlüsselspeicher verbunden. Sie können KMS-Schlüssel im benutzerdefinierten Schlüsselspeicher erstellen und verwenden.

  Der Unterstützungsschlüsselspeicher für einen AWS CloudHSM-Schlüsselspeicher ist der zugehörige AWS CloudHSM-Cluster. Der Unterstützungsschlüsselspeicher für einen externen Schlüsselspeicher ist der externe Schlüsselspeicher-Proxy und der externe Schlüsselmanager, den er unterstützt.

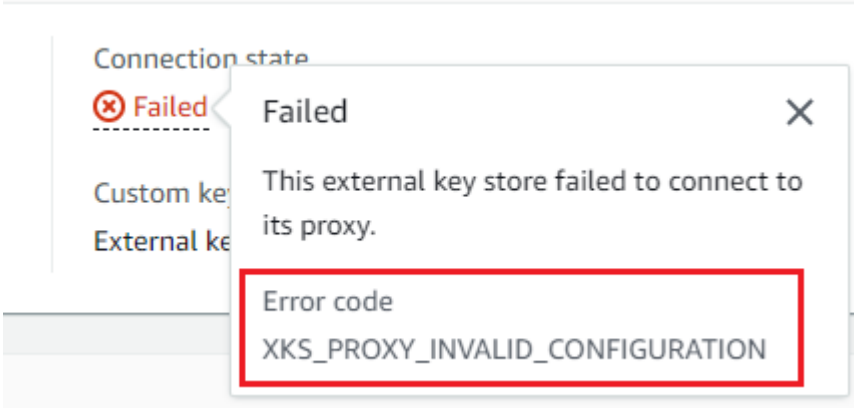
  Der Status „CONNECTED“ (VERBUNDEN) bedeutet, dass eine Verbindung erfolgreich war und der benutzerdefinierte Schlüsselspeicher nicht absichtlich getrennt wurde. Er bedeutet nicht, dass die Verbindung ordnungsgemäß funktioniert. Informationen zum Status des AWS CloudHSM Clusters, der Ihrem -AWS CloudHSMSchlüsselspeicher zugeordnet ist, finden Sie unter Abrufen von CloudWatch Metriken für AWS CloudHSM im AWS CloudHSM-Benutzerhandbuch. Informationen zum Status und Betrieb Ihres externen Schlüsselspeichers finden Sie in den Diagrammen im Abschnitt Überwachung auf der Detailseite für jeden externen Schlüsselspeicher. Details hierzu finden Sie unter Überwachung eines externen Schlüsselspeichers.
- `CONNECTING`: Der Prozess der Verbindung eines benutzerdefinierten Schlüsselspeichers ist im Gange. Dies ist ein vorübergehender Zustand.
- `DISCONNECTED`: Der benutzerdefinierte Schlüsselspeicher wurde noch nie mit seiner Unterstützung verbunden oder er wurde absichtlich mithilfe der AWS KMS Konsole oder der -DisconnectCustomKeyStoreOperation getrennt.
- `DISCONNECTING`: Der Prozess der Trennung eines benutzerdefinierten Schlüsselspeichers ist im Gange. Dies ist ein vorübergehender Zustand.
- `FAILED`: Ein Versuch, den benutzerdefinierten Schlüsselspeicher zu verbinden, ist fehlgeschlagen. Der `ConnectionErrorCode` in der DescribeCustomKeyStores Antwort weist auf das Problem hin.

Um einen benutzerdefinierten Schlüsselspeicher zu verbinden, muss sein Verbindungsstatus `DISCONNECTED` sein. Wenn der Verbindungsstatus `FAILED` lautet, identifizieren und lösen Sie das Problem anhand des `ConnectionErrorCode`. Trennen Sie dann den benutzerdefinierten Schlüsselspeicher, bevor Sie versuchen, die Verbindung wieder herzustellen. Hilfestellung bei

fehlgeschlagenen Verbindungen finden Sie unter Fehler bei der Verbindung mit dem externen Schlüsselspeicher. Hilfe beim Beantworten eines Verbindungsfehlercodes finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher.

Anzeigen des Verbindungsfehlercodes:

- Zeigen Sie in der DescribeCustomKeyStores Antwort den Wert des -`ConnectionErrorCode`Elements an. Die `DescribeCustomKeyStores`-Antwort enthält dieses Element nur, wenn der `ConnectionState` `FAILED` ist.
- Zum Anzeigen des Verbindungsfehlercodes in der AWS KMS-Konsole gehen Sie auf die Detailseite des externen Schlüsselspeichers und bewegen Sie den Mauszeiger über den Wert Failed (Fehlgeschlagen.



Verbinden eines externen Schlüsselspeichers (Konsole)

Sie können die AWS KMS-Konsole verwenden, um einen externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy zu verbinden.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Wählen Sie die Zeile des externen Schlüsselspeichers aus, den Sie verbinden möchten.

Wenn der Verbindungsstatus des externen Schlüsselspeichers FAILED (FEHLGESCHLAGEN) ist, müssen Sie die Verbindung des externen Schlüsselspeichers trennen, bevor Sie ihn verbinden.

5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Connect (Verbinden) aus.

Der Verbindungsvorgang dauert in der Regel etwa fünf Minuten. Wenn der Vorgang abgeschlossen ist, ändert sich der Verbindungsstatus in CONNECTED (VERBUNDEN).

Wenn der Verbindungsstatus Failed (Fehlgeschlagen) ist, bewegen Sie den Mauszeiger über den Verbindungsstatus, um den Verbindungsfehlercode zu sehen, der die Ursache des Fehlers erklärt. Hilfe beim Beantworten eines Verbindungsfehlercodes finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher. Um einen externen Schlüsselspeicher mit dem Verbindungsstatus Failed (Fehlgeschlagen) zu verbinden, müssen Sie zuerst den benutzerdefinierten Schlüsselspeicher trennen.

Verbinden eines externen Schlüsselspeichers (API)

Um eine Verbindung zu einem getrennten externen Schlüsselspeicher herzustellen, verwenden Sie die -ConnectCustomKeyStoreOperation.

Vor der Verbindung muss der Verbindungsstatus des externen Schlüsselspeichers `DISCONNECTED` sein. Wenn der Verbindungsstatus `FAILED` lautet, trennen Sie den externen Schlüsselspeicher und stellen Sie anschließend die Verbindung her.

Der Verbindungsvorgang dauert etwa fünf Minuten. Wenn er nicht schnell fehlschlägt, gibt `ConnectCustomKeyStore` eine HTTP 200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück. Diese erste Antwort gibt jedoch nicht an, dass die Verbindung erfolgreich war. Um festzustellen, ob der externe Schlüsselspeicher verbunden ist, sehen Sie sich den Verbindungsstatus in der DescribeCustomKeyStores Antwort an.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Identifizieren Sie den externen Schlüsselspeicher anhand der ID des benutzerdefinierten Schlüsselspeichers. Sie finden die ID auf der Seite Benutzerdefinierte Schlüsselspeicher in der -Konsole oder mithilfe der -DescribeCustomKeyStoresOperation. Ersetzen Sie vor Ausführung dieses Beispiels die Beispiel-ID durch eine gültige ID.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Der `ConnectCustomKeyStore`-Vorgang gibt den `ConnectionState` nicht in seiner Antwort zurück. Um zu überprüfen, ob der externe Schlüsselspeicher verbunden ist, verwenden Sie die -DescribeCustomKeyStoresOperation. Diese Produktion gibt standardmäßig alle benutzerdefinierten Schlüsselspeicher innerhalb des Kontos und der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die Antwort auf bestimmte benutzerdefinierte Schlüsselspeicher zu begrenzen. Der `ConnectionState`-Wert `CONNECTED` bedeutet, dass der externe Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy verbunden ist.

```
$ 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"
      }
    }
  ]
}
```

Wenn der `ConnectionState`-Wert in der `DescribeCustomKeyStores`-Antwort `FAILED` lautet, gibt das `ConnectionErrorCode`-Element den Grund für den Fehler an.

Im folgenden Beispiel bedeutet der Wert `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` für den `ConnectionErrorCode`, dass AWS KMS den VPC-Endpunkt-Service nicht finden kann, den es für die Kommunikation mit dem externen Schlüsselspeicher-Proxy verwendet. Stellen Sie sicher, dass der `XksProxyVpcEndpointServiceName` korrekt ist, dass der AWS KMS-Service-Prinzipal ein zulässiger Prinzipal für den VPC-Endpunkt-Service von Amazon ist und dass der VPC-Endpunkt-

Service die Annahme von Verbindungsanforderungen nicht erfordert. Hilfe beim Beantworten eines Verbindungsfehlercodes finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher.

```
$ 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"
      }
    }
  ]
}
```

Trennen eines externen Schlüsselspeichers (Konsole)

Sie können die AWS KMS-Konsole verwenden, um einen externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy zu verbinden. Dieser Vorgang dauert etwa 5 Minuten.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Wählen Sie die Zeile des externen Schlüsselspeichers aus, mit dem Sie die Verbindung trennen möchten.
5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Disconnect (Verbindung trennen) aus.

Nach Abschluss der Produktion ändert sich der Verbindungsstatus von DISCONNECTING (VERBINDUNG WIRD GETRENNT) in DISCONNECTED (VERBINDUNG GETRENNT). Wenn die Produktion fehlschlägt, wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehler bei der Verbindung mit dem externen Schlüsselspeicher.

Trennen eines externen Schlüsselspeichers (API)

Um die Verbindung eines verbundenen externen Schlüsselspeichers zu trennen, verwenden Sie die -DisconnectCustomKeyStoreOperation. Wenn die Produktion erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück. Der Vorgang dauert etwa fünf Minuten. Verwenden Sie die Operation , um den Verbindungsstatus des externen Schlüsselspeichers zu ermittelnDescribeCustomKeyStores.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

In diesem Beispiel wird die Verbindung eines externen Schlüsselspeichers mit Konnektivität eines VPC-Endpunkt-Service getrennt. Vor der Ausführung dieses Beispiels müssen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers durch eine gültige ID ersetzen.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Um zu überprüfen, ob der externe Schlüsselspeicher getrennt ist, verwenden Sie die -DescribeCustomKeyStoresOperation. Diese Produktion gibt standardmäßig alle benutzerdefinierten Schlüsselspeicher innerhalb des Kontos und der Region zurück. Sie können jedoch entweder den Parameter `CustomKeyStoreId` oder `CustomKeyStoreName` (aber nicht beide) verwenden, um die Antwort auf bestimmte benutzerdefinierte Schlüsselspeicher zu begrenzen. Der `ConnectionState`-Wert `DISCONNECTED` bedeutet, dass in diesem Beispiel der externe Schlüsselspeicher nicht mehr mit seinem externen Schlüsselspeicher-Proxy verbunden ist.

```
$ 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"
      }
    }
  ]
}
```

Löschen eines externen Schlüsselspeichers

Wenn Sie einen externen Schlüsselspeicher löschen, löscht AWS KMS alle Metadaten über den externen Schlüsselspeicher aus AWS KMS, einschließlich der Informationen über seinen externen Schlüsselspeicher-Proxy. Dieser Vorgang wirkt sich nicht auf den Proxy des externen Schlüsselspeichers, den externen Schlüsselmanager, externe Schlüssel oder AWS-Ressourcen aus, die Sie zur Unterstützung des externen Schlüsselspeichers erstellt haben, wie z. B. eine Amazon VPC oder einen VPC-Endpunkt-Service.

Bevor Sie einen externen Schlüsselspeicher löschen, müssen Sie alle KMS-Schlüssel aus dem Schlüsselspeicher löschen und die Verbindung des Schlüsselspeichers mit seinem externen Schlüsselspeicher-Proxy trennen. Andernfalls schlagen Versuche, den Schlüsselspeicher zu löschen, fehl.

Das Löschen eines externen Schlüsselspeichers ist nicht rückgängig zu machen, aber Sie können einen neuen externen Schlüsselspeicher erstellen und ihn mit demselben externen Schlüsselspeicher-Proxy und externen Schlüsselmanager verknüpfen. Allerdings können Sie die KMS-Schlüssel mit symmetrischer Verschlüsselung im externen Schlüsselspeicher nicht erneut erstellen, selbst wenn Sie Zugriff auf dasselbe externe Schlüsselmaterial haben. AWS KMS enthält Metadaten im symmetrischen Geheimtext, die für jeden KMS-Schlüssel eindeutig sind. Diese Sicherheitsfunktion sorgt dafür, dass nur der KMS-Schlüssel, der die Daten verschlüsselt hat, sie entschlüsseln kann.

Anstatt den externen Schlüsselspeicher zu löschen, sollten Sie dessen Verbindung trennen. Während ein externer Schlüsselspeicher getrennt ist, können Sie den externen Schlüsselspeicher und seine AWS KMS keys-Schlüssel verwalten, aber Sie können keine KMS-Schlüssel im externen Schlüsselspeicher erstellen oder verwenden. Sie können den externen Schlüsselspeicher jederzeit wieder verbinden und seine KMS-Schlüssel zum Ver- und Entschlüsseln von Daten verwenden.

Für einen nicht verbundenen externen Schlüsselspeicher-Proxy oder seine nicht verfügbaren KMS-Schlüssel fallen keine Kosten an.

Themen

- Löschen eines externen Schlüsselsspeichers (Konsole)
- Löschen eines externen Schlüsselsspeichers (API)

Löschen eines externen Schlüsselsspeichers (Konsole)

Sie können die AWS KMS-Konsole verwenden, um einen externen Schlüsselspeicher zu löschen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Suchen Sie die Zeile, die den externen Schlüsselspeicher darstellt, den Sie löschen möchten. Wenn der Verbindungsstatus des externen Schlüsselspeichers nicht DISCONNECTED (VERBINDUNG GETRENNT) ist, müssen Sie die Verbindung des externen Schlüsselspeichers trennen, bevor Sie ihn löschen.
5. Wählen Sie im Menü Key store actions (Schlüsselspeicheraktionen) die Option Delete (Löschen) aus.

Nach Abschluss des Vorgangs wird eine Erfolgsmeldung angezeigt und der externe Schlüsselspeicher wird nicht mehr in der Schlüsselspeicherliste angezeigt. Wenn die Produktion nicht erfolgreich ist, wird eine Fehlermeldung mit einer Beschreibung des Problems und Hilfestellung zur Fehlerbehebung angezeigt. Wenn Sie weitere Hilfe benötigen, beachten Sie den Abschnitt Fehlerbehebung bei externen Schlüsselspeichern.

Löschen eines externen Schlüsselsspeichers (API)

Um einen externen Schlüsselspeicher zu löschen, verwenden Sie die -DeleteCustomKeyStoreOperation. Wenn die Produktion erfolgreich ausgeführt wurde, gibt AWS KMS eine HTTP-200-Antwort und ein JSON-Objekt ohne Eigenschaften zurück.

Trennen Sie zunächst die Verbindung des externen Schlüsselspeichers. Vor der Ausführung dieses Befehls müssen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers durch eine gültige ID ersetzen.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Nachdem der externe Schlüsselspeicher getrennt wurde, können Sie den DeleteCustomKeyStore Vorgang verwenden, um ihn zu löschen.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Um zu bestätigen, dass der externe Schlüsselspeicher gelöscht wurde, verwenden Sie die -DescribeCustomKeyStoresOperation.

```
$ aws kms describe-custom-key-stores

{
    "CustomKeyStores": []
}
```

Wenn Sie den Namen oder die ID eines benutzerdefinierten Schlüsselspeichers angeben, der nicht mehr vorhanden ist, gibt AWS KMS die `CustomKeyStoreNotFoundException`-Ausnahme zurück.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0

An error occurred (CustomKeyStoreNotFoundException) when calling the
 DescribeCustomKeyStore operation:
```

Verwalten von KMS-Schlüsseln in einem externen Schlüsselspeicher

Zum Erstellen, Anzeigen, Verwalten, Verwenden und Planen der Löschung von KMS-Schlüsseln in einem externen Schlüsselspeicher verwenden Sie ganz ähnliche Verfahren wie für andere KMS-Schlüssel. Wenn Sie jedoch einen KMS-Schlüssel in einem externen Schlüsselspeicher erstellen, geben Sie einen externen Schlüsselspeicher und einen externen Schlüssel an. Wenn Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher verwenden, werden die Ver- und Entschlüsselungsvorgänge von Ihrem externen Schlüsselmanager unter Verwendung des angegebenen externen Schlüssels durchgeführt.

AWS KMS kann keine kryptografischen Schlüssel in Ihrem externen Schlüsselmanager erstellen, anzeigen, aktualisieren oder löschen. AWS KMS greift niemals direkt auf Ihren externen Schlüsselmanager oder einen externen Schlüssel zu. Alle Anforderungen für kryptografische Vorgänge werden vom Proxy Ihres externen Schlüsselspeichers vermittelt. Zur Verwendung eines KMS-Schlüssels in einem externen Schlüsselspeicher muss der externe Schlüsselspeicher, der den KMS-Schlüssel hostet, mit seinem externen Schlüsselspeicher-Proxy verbunden sein.

Unterstützte Funktionen

Neben den in diesem Abschnitt beschriebenen Verfahren können Sie mit KMS-Schlüsseln in einem externen Schlüsselspeicher folgende Aktionen ausführen:

- Verwenden von Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen zur Steuerung des Zugriffs auf die KMS-Schlüssel
- Aktivieren und Deaktivieren der KMS-Schlüssel. Diese Aktionen wirken sich nicht auf den externen Schlüssel in Ihrem externen Schlüsselmanager aus.
- Zuweisen von Tags und Erstellen von Aliassen und Autorisieren des Zugriffs auf die KMS-Schlüssel mithilfe der attributbasierten Zugriffskontrolle (ABAC)
- Verwenden der KMS-Schlüssel mit AWS-Services-Services, die in AWS KMS integriert werden können und Unterstützen von kundenverwalteten KMS-Schlüsseln

Nicht unterstützte Funktionen

- Externe Schlüsselspeicher unterstützen nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können keine HMAC-KMS-Schlüssel oder asymmetrische KMS-Schlüssel in einem externen Schlüsselspeicher erstellen.
- GenerateDataKeyPair und GenerateDataKeyPairWithoutPlaintext werden auf KMS-Schlüsseln in einem externen Schlüsselspeicher nicht unterstützt.
- Sie können keine AWS CloudFormation-Vorlage verwenden, um einen externen Schlüsselspeicher oder einen KMS-Schlüssel in einem externen Schlüsselspeicher zu erstellen.
- Multiregionale Schlüssel werden in einem externen Schlüsselspeicher nicht unterstützt.
- KMS-Schlüssel mit importiertem Schlüsselmaterial werden in einem externen Schlüsselspeicher nicht unterstützt.
- Automatische Schlüsselrotation wird für KMS-Schlüssel in einen externen Schlüsselspeicher nicht unterstützt.

Themen

Erstellen von KMS-Schlüsseln in einem externen Schlüsselspeicher

Nachdem Sie Ihren externen Schlüsselspeicher erstellt und verbunden haben, können Sie AWS KMS keys in Ihrem Schlüsselspeicher erstellen. Dabei muss es sich um KMS-Schlüssel mit symmetrischer Verschlüsselung mit dem Ursprungswert External key store (Externer Schlüsselspeicher) (`EXTERNAL_KEY_STORE`) handeln. Sie können asymmetrische KMS-Schlüssel, HMAC-KMS-Schlüssel oder KMS-Schlüssel mit importiertem Schlüsselmaterial nicht in einem benutzerdefinierten Schlüsselspeicher erstellen. Außerdem können Sie auch keine KMS-Schlüssel mit symmetrischer Verschlüsselung in einem benutzerdefinierten Schlüsselspeicher verwenden, um asymmetrische Datenschlüsselpaare zu generieren.

Ein KMS-Schlüssel in einem externen Schlüsselspeicher hat möglicherweise eine schlechtere Latenz, Haltbarkeit und Verfügbarkeit als ein Standard-KMS-Schlüssel, da er von Komponenten abhängt, die sich außerhalb von AWS befinden. Bevor Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher erstellen oder verwenden, prüfen Sie, ob Sie einen Schlüssel mit den Eigenschaften eines externen Schlüsselspeichers benötigen.

> **Note**
>
> Einige externe Schlüsselmanager bieten eine einfachere Methode zum Erstellen von KMS-Schlüsseln in einem externen Schlüsselspeicher. Weitere Informationen finden Sie in der Dokumentation zum externen Schlüsselmanager.

Zum Erstellen eines KMS-Schlüssels in Ihrem externen Schlüsselspeicher müssen Sie Folgendes angeben:

- Die ID Ihres externen Schlüsselspeichers.
- Externer Schlüsselspeicher (`EXTERNAL_KEY_STORE`) als Herkunft des Schlüsselmaterials.
- Die ID eines vorhandenen externen Schlüssels im externen Schlüsselmanager, der Ihrem externen Schlüsselspeicher zugeordnet ist. Dieser externe Schlüssel dient als Schlüsselmaterial für den

KMS-Schlüssel. Sie können die ID des externen Schlüssels nicht mehr ändern, nachdem Sie den KMS-Schlüssel erstellt haben.

AWS KMS stellt die ID des externen Schlüssels dem Proxy Ihres externen Schlüsselspeichers in Anforderungen für Verschlüsselungs- und Entschlüsselungsvorgänge zur Verfügung. AWS KMS kann nicht direkt auf Ihren externen Schlüsselmanager oder einen seiner kryptografischen Schlüssel zugreifen.

Zusätzlich zum externen Schlüssel enthält ein KMS-Schlüssel in einem externen Schlüsselspeicher auch AWS KMS-Schlüsselmaterial. Alle unter dem KMS-Schlüssel verschlüsselten Daten werden zunächst in AWS KMS mit dem AWS KMS-Schlüsselmaterial des Schlüssels verschlüsselt und dann von Ihrem externen Schlüsselmanager mit Ihrem externen Schlüssel. Dieser Prozess der doppelten Verschlüsselung sorgt dafür, dass der durch einen KMS-Schlüssel in einem externen Schlüsselspeicher geschützte Geheimtext mindestens so stark ist wie nur durch AWS KMS geschützter Geheimtext. Details hierzu finden Sie unter Funktionsweise externer Schlüsselspeicher.

Wenn der `CreateKey`-Vorgang erfolgreich ist, lautet der Schlüsselstatus des neuen KMS-Schlüssels `Enabled`. Wenn Sie einen KMS-Schlüssel in einem externen Schlüsselspeicher anzeigen, können Sie typische Eigenschaften wie die Schlüssel-ID, die Schlüsselspezifikation, die Schlüsselnutzung, den Schlüsselstatus und das Erstellungsdatum sehen. Sie können aber auch die ID und den Verbindungsstatus des externen Schlüsselspeichers sowie die ID des externen Schlüssels sehen.

Wenn Ihr Versuch, einen KMS-Schlüssel in Ihrem externen Schlüsselspeicher zu erstellen, fehlschlägt, ermitteln Sie die Ursache anhand der Fehlermeldung. Sie kann darauf hinweisen, dass der externe Schlüsselspeicher nicht verbunden ist (`CustomKeyStoreInvalidStateException`), dass der Proxy Ihres externen Schlüsselspeichers keinen externen Schlüssel mit der angegebenen externen Schlüssel-ID finden kann (`XksKeyNotFoundException`) oder dass der externe Schlüssel bereits mit einem KMS-Schlüssel im selben externen Schlüsselspeicher verknüpft ist (`XksKeyAlreadyInUseException`).

.

Ein Beispiel des AWS CloudTrail-Protokolls des Vorgang, der einen KMS-Schlüssel in einem externen Schlüsselspeicher erstellt, finden Sie unter CreateKey.

Themen

- Anforderungen an einen KMS-Schlüssel in einem externen Schlüsselspeicher
- Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher (Konsole)

- Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher (AWS KMS-API)

Anforderungen an einen KMS-Schlüssel in einem externen Schlüsselspeicher

Zum Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher sind die folgenden Eigenschaften des externen Schlüsselspeichers, des KMS-Schlüssels und des externen Schlüssels, der als externes kryptografisches Schlüsselmaterial für den KMS-Schlüssel dient, erforderlich.

Anforderungen an einen externen Schlüsselspeicher

- Muss mit seinem externen Schlüsselspeicher-Proxy verbunden sein.

  Informationen zum Verbindungsstatus Ihres externen Schlüsselspeichers finden Sie unter Anzeigen eines externen Schlüsselspeichers. Informationen zum Verbinden Ihres externen Schlüsselspeichers finden Sie unter Herstellen und Trennen der Verbindung eines externen Schlüsselspeichers.

Anforderungen an KMS-Schlüssel

Diese Eigenschaften können nach dem Erstellen des KMS-Schlüssels nicht mehr geändert werden.

- Schlüsselspezifikation: SYMMMETRIC_DEFAULT
- Schlüsselnutzung: ENCRYPT_DECRYPT
- Schlüsselmaterialursprung: EXTERNAL_KEY_STORE
- Multi-Region: FALSE

Anforderungen an externe Schlüssel

- Kryptografischer 256-Bit-AES-Schlüssel (256 zufällige Bits). Die `KeySpec` des externen Schlüssels muss `AES_256` sein.
- Aktiviert und zur Verwendung verfügbar. Die `Status` des externen Schlüssels muss `ENABLED` sein.
- Konfiguriert für Verschlüsselung und Entschlüsselung. Die `KeyUsage` des externen Schlüssels muss `ENCRYPT` und `DECRYPT` enthalten.
- Wird nur mit diesem KMS-Schlüssel verwendet. Jeder `KMS key` in einem externen Schlüsselspeicher muss mit einem anderen externen Schlüssel verbunden sein.

AWS KMS empfiehlt außerdem, den externen Schlüssel ausschließlich für den externen Schlüsselspeicher zu verwenden. Diese Einschränkung macht es einfacher, Probleme mit dem Schlüssel zu erkennen und zu beheben.

- Zugriff durch den Proxy des externen Schlüsselspeichers für den externen Schlüsselspeicher möglich.

  Wenn der Proxy des externen Schlüsselspeichers den Schlüssel mit der angegebenen externen Schlüssel-ID nicht finden kann, schlägt der Vorgang `CreateKey` fehl.
- Kann den erwarteten Datenverkehr bewältigen, den Ihre Nutzung von AWS-Services erzeugt. AWS KMS empfiehlt, dass externe Schlüssel auf die Verarbeitung von bis zu 1 800 Anforderungen pro Sekunde vorbereitet sind.

Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher (Konsole)

Es gibt zwei Möglichkeiten, einen KMS-Schlüssel in einem externen Schlüsselspeicher zu erstellen.

- Methode 1 (empfohlen): Wählen Sie einen externen Schlüsselspeicher und erstellen Sie dann einen KMS-Schlüssel in diesem externen Schlüsselspeicher.
- Methode 2: Erstellen Sie einen KMS-Schlüssel und geben Sie dann an, dass er sich in einem externen Schlüsselspeicher befindet.

Wenn Sie Methode 1 verwenden, bei der Sie Ihren externen Schlüsselspeicher auswählen, bevor Sie Ihren Schlüssel erstellen, wählt AWS KMS alle erforderlichen KMS-Schlüsseleigenschaften für Sie aus und trägt die ID Ihres externen Schlüsselspeichers ein. Diese Methode vermeidet Fehler, die Ihnen bei der Erstellung Ihres KMS-Schlüssels unterlaufen könnten.

> **Note**
>
> Nehmen Sie keine vertraulichen oder sensiblen Informationen in den Alias, in der Beschreibung oder in den Tags auf. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Methode 1 (empfohlen): Starten in Ihrem externen Schlüsselspeicher

Um diese Methode zu verwenden, wählen Sie Ihren externen Schlüsselspeicher und erstellen dann einen KMS-Schlüssel. Die AWS KMS-Konsole wählt alle erforderlichen Eigenschaften für Sie aus und

trägt die ID Ihres externen Schlüsselspeichers ein. Diese Methode vermeidet viele Fehler, die Ihnen bei der Erstellung Ihres KMS-Schlüssels unterlaufen könnten.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie im Navigationsbereich Custom key stores (Benutzerdefinierte Schlüsselspeicher), External key stores (Externe Schlüsselspeicher) aus.
4. Wählen Sie den Namen Ihres externen Schlüsselspeichers aus.
5. Wählen Sie oben rechts die Option Create a KMS key in this key store (Erstellen eines KMS-Schlüssels in diesem Schlüsselspeicher) aus.

   Wenn der externe Schlüsselspeicher nicht verbunden ist, werden Sie aufgefordert, ihn zu verbinden. Wenn der Verbindungsversuch fehlschlägt, müssen Sie das Problem lösen und den externen Schlüsselspeicher verbinden, bevor Sie darin einen neuen KMS-Schlüssel erstellen können.

   Wenn der externe Schlüsselspeicher verbunden ist, werden Sie zur Seite Customer managed keys (Kundenverwaltete Schlüssel) weitergeleitet, auf der Sie einen Schlüssel erstellen können. Die erforderlichen Schlüsselkonfigurationswerte wurden bereits für Sie ausgewählt. Außerdem ist die benutzerdefinierte Schlüsselspeicher-ID Ihres externen Schlüsselspeichers ausgefüllt, Sie können diese aber ändern.
6. Geben Sie die Schlüssel-ID eines externen Schlüssels in Ihren externen Schlüsselmanager ein. Dieser externe Schlüssel muss die Anforderungen für die Verwendung mit einem KMS-Schlüssel erfüllen. Sie können diesen Wert nach der Erstellung des KMS-Schlüssels nicht mehr ändern.

   Wenn der externe Schlüssel mehrere IDs hat, geben Sie die Schlüssel-ID ein, die der externe Schlüsselspeicher-Proxy zur Identifizierung des externen Schlüssels verwendet.
7. Bestätigen Sie, dass Sie beabsichtigen, einen KMS-Schlüssel im angegebenen externen Schlüsselspeicher zu erstellen.
8. Wählen Sie Weiter aus.

   Der Rest dieses Verfahrens entspricht dem Erstellen eines Standard-KMS-Schlüssels.
9. Geben Sie einen Alias (erforderlich) und eine Beschreibung (optional) für den KMS-Schlüssel ein.

10. (Optional). Fügen Sie auf der Seite Add Tags (Tags hinzufügen) Tags hinzu, um Ihre KMS-Schlüssel identifizieren zu können und sie zu kategorisieren.

    Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.
11. Wählen Sie Weiter aus.
12. Sie können im Abschnitt Key administrators (Schlüsseladministratoren) die IAM-Benutzer und -Rollen auswählen, die den KMS-Schlüssel verwalten dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsseladministratoren die Verwaltung des KMS-Schlüssels.

    > **Note**
    >
    > Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssel auch über IAM-Richtlinien erhalten.
    > Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.
13. (Optional) Um zu verhindern, dass diese Schlüsseladministratoren diesen KMS-Schlüssel löschen, deaktivieren Sie das Kontrollkästchen Allow key administrators to delete this key (Ermöglicht Schlüsseladministratoren das Löschen dieses Schlüssels).

    Das Löschen eines KMS-Schlüssels ist ein endgültiger und irreversibler Vorgang, der dazu führen kann, dass Geheimtext nicht wiederhergestellt werden kann. Sie können einen symmetrischen KMS-Schlüssel in einem externen Schlüsselspeicher nicht wiederherstellen, selbst wenn Sie über das externe Schlüsselmaterial verfügen. Das Löschen eines KMS-Schlüssels hat jedoch keine Auswirkungen auf den zugehörigen externen Schlüssel. Informationen zum Löschen eines KMS-Schlüssels aus einem externen Schlüsselspeicher finden Sie unter Planen der Löschung von KMS-Schlüsseln aus einem externen Schlüsselspeicher.
14. Wählen Sie Weiter aus.
15. Wählen Sie im Abschnitt This account (dieses Konto) die IAM-Benutzer und -Rollen in diesem AWS-Konto aus, die den KMS-Schlüssel für kryptografische Operationen verwenden dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssels.

> **Note**
>
> Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssels auch über IAM-Richtlinien erhalten.
> Bewährte IAM-Methoden raten von der Verwendung von IAM-Benutzern mit langfristigen Anmeldeinformationen ab. Verwenden Sie nach Möglichkeit IAM-Rollen, die temporäre Anmeldeinformationen bereitstellen. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

16. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other (Andere Konten) AWS-Konten unten auf der Seite die Option Add another (Weiteres Konto hinzufügen)AWS-Konto aus und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

> **Note**
>
> Administratoren der anderen AWS-Konten müssen auch Zugriff auf den KMS-Schlüssel erlauben, indem Sie IAM-Richtlinien für ihre Benutzer erstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

17. Wählen Sie Weiter.
18. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
19. Wählen Sie danach Finish (Fertigstellen) aus.

Methode 2: Starten in kundenverwalteten Schlüsseln

Dieses Verfahren entspricht dem Verfahren zum Erstellen eines symmetrischen Verschlüsselungsschlüssels mit AWS KMS-Schlüsselmaterial. In diesem Verfahren geben Sie jedoch die benutzerdefinierte Schlüsselspeicher-ID des externen Schlüsselspeichers und die Schlüssel-ID des externen Schlüssels an. Sie müssen auch die erforderlichen Eigenschaftswerte für einen KMS-Schlüssel in einem externen Schlüsselspeicher angeben, z. B. die Schlüsselspezifikation und die Schlüsselnutzung.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.
2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Klicken Sie auf Create key.
5. Wählen Sie Symmetric (Symmetrisch).
6. Unter Key usage (Schlüsselverwendung) ist die Option Encrypt and decrypt (Verschlüsseln und Entschlüsseln) für Sie ausgewählt. Ändern Sie dies nicht.
7. Wählen Sie Advanced options (Erweiterte Optionen) aus.
8. Wählen Sie unter Key material origin (Schlüsselmaterialursprung) die Option External key store (Externer Schlüsselspeicher) aus.
9. Bestätigen Sie, dass Sie beabsichtigen, einen KMS-Schlüssel im angegebenen externen Schlüsselspeicher zu erstellen.
10. Wählen Sie Weiter aus.
11. Wählen Sie die Zeile, die den externen Schlüsselspeicher für den neuen KMS-Schlüssel darstellt.

    Sie können keinen getrennten externen Schlüsselspeicher auswählen. Zum Verbinden eines nicht verbundenen Schlüsselspeichers wählen Sie den Namen des Schlüsselspeichers und dann unter Key store actions (Schlüsselspeicheraktionen) die Option Connect (Verbinden) aus. Details hierzu finden Sie unter Verbinden eines externen Schlüsselspeichers (Konsole).
12. Geben Sie die Schlüssel-ID eines externen Schlüssels in Ihren externen Schlüsselmanager ein. Dieser externe Schlüssel muss die Anforderungen für die Verwendung mit einem KMS-Schlüssel erfüllen. Sie können diesen Wert nach der Erstellung des KMS-Schlüssels nicht mehr ändern.

    Wenn der externe Schlüssel mehrere IDs hat, geben Sie die Schlüssel-ID ein, die der externe Schlüsselspeicher-Proxy zur Identifizierung des externen Schlüssels verwendet.
13. Wählen Sie Weiter aus.

    Der Rest dieses Verfahrens entspricht dem Erstellen eines Standard-KMS-Schlüssels.
14. Geben Sie einen Alias und eine optionale Beschreibung für den KMS-Schlüssel ein.
15. (Optional). Fügen Sie auf der Seite Add Tags (Tags hinzufügen) Tags hinzu, um Ihre KMS-Schlüssel identifizieren zu können und sie zu kategorisieren.

Wenn Sie Tags auf AWS-Ressourcen anwenden, erzeugt AWS einen Kostenzuordnungsbericht mit Nutzungs- und Kostendaten der Tags. Markierungen können auch verwendet werden, um den Zugriff auf einen KMS-Schlüssel zu steuern. Weitere Informationen über das Markieren von KMS-Schlüsseln finden Sie unter Tagging von Schlüsseln und ABAC für AWS KMS.

16. Wählen Sie Weiter aus.
17. Sie können im Abschnitt Key administrators (Schlüsseladministratoren) die IAM-Benutzer und -Rollen auswählen, die den KMS-Schlüssel verwalten dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsseladministratoren die Verwaltung des KMS-Schlüssels.

> **Note**
>
> Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssel auch über IAM-Richtlinien erhalten.

18. (Optional) Um zu verhindern, dass diese Schlüsseladministratoren diesen KMS-Schlüssel löschen, deaktivieren Sie das Kontrollkästchen Allow key administrators to delete this key (Ermöglicht Schlüsseladministratoren das Löschen dieses Schlüssels).

    Das Löschen eines KMS-Schlüssels ist ein endgültiger und irreversibler Vorgang, der dazu führen kann, dass Geheimtext nicht wiederhergestellt werden kann. Sie können einen symmetrischen KMS-Schlüssel in einem externen Schlüsselspeicher nicht wiederherstellen, selbst wenn Sie über das externe Schlüsselmaterial verfügen. Das Löschen eines KMS-Schlüssels hat jedoch keine Auswirkungen auf den zugehörigen externen Schlüssel. Informationen zum Löschen eines KMS-Schlüssels aus einem externen Schlüsselspeicher finden Sie unter Planen der Löschung von KMS-Schlüsseln aus einem externen Schlüsselspeicher.
19. Wählen Sie Weiter aus.
20. Wählen Sie im Abschnitt This account (dieses Konto) die IAM-Benutzer und -Rollen in diesem AWS-Konto aus, die den KMS-Schlüssel für kryptografische Operationen verwenden dürfen. Weitere Informationen finden Sie unter Erlaubt Schlüsselbenutzern die Verwendung des KMS-Schlüssels.

> **Note**
>
> Daneben können IAM-Benutzer und -Rollen die erforderlichen Berechtigungen zur Verwendung des KMS-Schlüssels auch über IAM-Richtlinien erhalten.

21. (Optional) Sie können anderen AWS-Konten erlauben, diesen KMS-Schlüssel für kryptografische Operationen zu verwenden. Wählen Sie dazu im Abschnitt Other (Andere Konten) AWS-Konten unten auf der Seite die Option Add another (Weiteres Konto hinzufügen)AWS-Konto aus und geben Sie die AWS-Konto-ID eines externen Kontos ein. Wiederholen Sie diesen Schritt, um weitere externe Konten hinzufügen.

    > **Note**
    >
    > Administratoren der anderen AWS-Konten müssen auch Zugriff auf den KMS-Schlüssel erlauben, indem Sie IAM-Richtlinien für ihre Benutzer erstellen. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung des KMS-Schlüssels erlauben.

22. Wählen Sie Weiter.
23. Überprüfen Sie die gewählten Einstellungen. Sie können immer noch zurückgehen und alle Einstellungen ändern.
24. Wählen Sie danach Finish (Fertigstellen) aus.

Wenn der Vorgang erfolgreich abgeschlossen wird, wird der neue KMS-Schlüssel in dem ausgewählten externen Schlüsselspeicher angezeigt. Wenn Sie den Namen oder Alias des neuen KMS-Schlüssels auswählen, werden auf der Registerkarte Cryptographic configuration (Kryptografische Konfiguration) auf der Detailseite der Ursprung des KMS-Schlüssels (Externer Schlüsselspeicher), der Name, die ID und der Typ des benutzerdefinierten Schlüsselspeichers sowie die ID, die Schlüsselnutzung und der Status des externen Schlüssels angezeigt. Wenn der Vorgang fehlschlägt, wird unter eine Fehlermeldung mit einer Beschreibung des Fehlers angezeigt. Informationen finden Sie unter Fehlerbehebung bei externen Schlüsselspeichern.

> **Tip**
>
> Um die Identifizierung von KMS-Schlüsseln in einem benutzerdefinierten Schlüsselspeicher zu erleichtern, fügen Sie auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) die Spalten Origin (Urspring) und Custom key store ID (ID des benutzerdefinierten Schlüsselspeichers) zur Anzeige hinzu. Zum Ändern der Tabellenfelder wählen Sie das Zahnradsymbol rechts oben auf der Seite aus. Details hierzu finden Sie unter Anpassen Ihrer KMS-Schlüsseltabellen.

Erstellen eines KMS-Schlüssels in einem externen Schlüsselspeicher (AWS KMS-API)

Um einen neuen KMS-Schlüssel in einem externen Schlüsselspeicher zu erstellen, verwenden Sie die -CreateKeyOperation. Die folgenden Parameter sind erforderlich:

- Der `Origin`-Wert muss `EXTERNAL_KEY_STORE` lauten.
- Der `CustomKeyStoreId`-Parameter identifiziert Ihren externen Schlüsselspeicher. Der `ConnectionState` des angegebenen externen Schlüsselspeichers muss `CONNECTED` sein. Verwenden Sie den `DescribeCustomKeyStores`-Vorgang um die `CustomKeyStoreId` und den `ConnectionState` zu ermitteln.
- Der `XksKeyId`-Parameter identifiziert den externen Schlüssel. Dieser externe Schlüssel muss die Anforderungen für die Zuordnung zu einem KMS-Schlüssel erfüllen.

Sie können auch jeden der optionalen Parameter des `CreateKey`-Vorgangs verwenden, z. B. die `Policy`- oder Tags-Parameter.

> **Note**
>
> Geben Sie keine vertraulichen oder sensiblen Informationen in die Felder `Description` oder `Tags` ein. Diese Felder können in CloudTrail Protokollen und anderen Ausgaben im Klartext vorkommen.

Für diese Beispiele wird die AWS Command Line Interface (AWS CLI) verwendet. Sie können aber jede unterstützte Programmiersprache nutzen.

Dieser Beispielbefehl verwendet die -CreateKeyOperation, um einen KMS-Schlüssel in einem externen Schlüsselspeicher zu erstellen. Die Antwort umfasst die Eigenschaften der KMS-Schlüssel, die ID des externen Schlüsselspeichers sowie ID, Nutzung und Status des externen Schlüssels. Detaillierte Informationen zu diesen Feldern finden Sie unter Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher.

Wenn Sie diesen Befehl ausführen, denken Sie daran, die ID des benutzerdefinierten Schlüsselspeichers in dem Beispiel durch eine gültige ID zu ersetzen.

```
$ 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"
    }
  }
}
```

Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher

Um die KMS-Schlüssel in einem externen Schlüsselspeicher anzuzeigen, verwenden Sie die -AWS KMSKonsole oder die -DescribeKeyOperation. Sie können dieselben Vorgehensweisen anwenden, die Sie auch für die Anzeige aller kundenverwalteten AWS KMS-Schlüssel verwenden würden. Informationen zu den Grundlagen finden Sie unter Anzeigen von Schlüsseln.

In der AWS KMS-Konsole werden die KMS-Schlüssel in Ihrem externen Schlüsselspeicher auf der Seite Customer managed keys (kundenverwaltete Schlüssel) zusammen mit allen anderen kundenverwalteten Schlüsseln in Ihrem AWS-Konto-Konto und Ihrer Region angezeigt. Um KMS-Schlüssel in einem externen Schlüsselspeicher zu identifizieren, filtern Sie nach dem eindeutigen Ursprungswert, dem externen Schlüsselspeicher und der ID des benutzerdefinierten Schlüsselspeichers.

Weitere Informationen finden Sie unter Anzeigen eines externen Schlüsselspeichers, Überwachung eines externen Schlüsselspeichers und AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

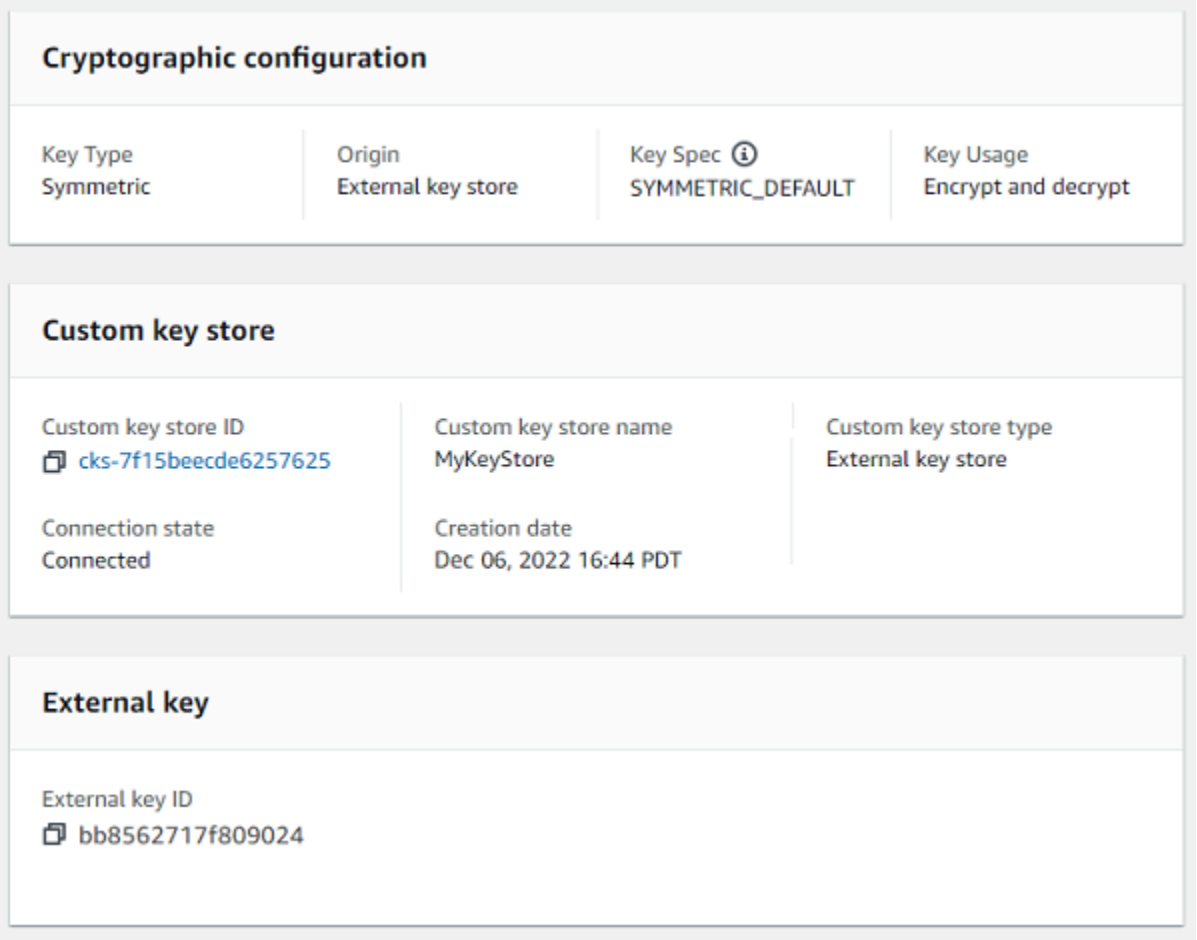
Themen

- Eigenschaften von KMS-Schlüsseln in einem externen Schlüsselspeicher
- Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher (Konsole)
- Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher (AWS KMS-API)

Eigenschaften von KMS-Schlüsseln in einem externen Schlüsselspeicher

Wie alle KMS-Schlüssel haben auch die KMS-Schlüssel in einem externen Schlüsselspeicher einen Schlüssel-ARN, eine Schlüsselspezifikation und Schlüsselnutzungswerte, aber sie haben auch Eigenschaften und Eigenschaftswerte, die spezifisch für KMS-Schlüssel in einem externen Schlüsselspeicher sind. Der Ursprungswert für alle KMS-Schlüssel in externen Schlüsselspeichern ist zum Beispiel External key store (Externer Schlüsselspeicher).

Für einen KMS-Schlüssel in einem externen Schlüsselspeicher enthält die Registerkarte Cryptographic configuration (Kryptografische Konfiguration) in der AWS KMS-Konsole zwei zusätzliche Abschnitte: Custom key store (Benutzerdefinierter Schlüsselspeicher) und External key (Externer Schlüssel).



Eigenschaften eines benutzerdefinierten Schlüsselspeichers

Die folgenden Werte werden im Abschnitt Benutzerdefinierter Schlüsselspeicher der Registerkarte Kryptografische Konfiguration und in der DescribeKey Antwort angezeigt. Diese Eigenschaften gelten für alle benutzerdefinierten Schlüsselspeicher, einschließlich AWS CloudHSM-Schlüsselspeicher und externe Schlüsselspeicher.

ID des benutzerdefinierten Schlüsselspeichers

Eine eindeutige ID, die AWS KMS dem benutzerdefinierten Schlüsselspeicher zuweist

Name des benutzerdefinierten Schlüsselspeichers

Ein Anzeigename, den Sie dem benutzerdefinierten Schlüsselspeicher bei dessen Erstellung zuweisen. Sie können diesen Wert jederzeit ändern.

Typ des benutzerdefinierten Schlüsselspeichers

Der Typ des benutzerdefinierten Schlüsselspeichers. Gültige Werte sind AWS CloudHSM (`AWS_CLOUDHSM`) oder „Externer Schlüsselspeicher“ (`EXTERNAL_KEY_STORE`). Der Typ kann nach der Erstellung des benutzerdefinierten Schlüsselspeichers nicht mehr geändert werden.

Erstellungsdatum

Das Datum, an dem der benutzerdefinierte Schlüsselspeicher erstellt wurde. Dieses Datum wird in Ortszeit für die AWS-Region angezeigt.

Verbindungsstatus

Zeigt an, ob der benutzerdefinierte Schlüsselspeicher mit dem Unterstützungsschlüsselspeicher verbunden ist. Der Verbindungsstatus ist nur dann `DISCONNECTED`, wenn der benutzerdefinierte Schlüsselspeicher noch nie mit dem Unterstützungsschlüsselspeicher verbunden war oder die Verbindung absichtlich getrennt wurde. Details hierzu finden Sie unter the section called “Verbindungsstatus”.

Eigenschaften des externen Schlüssels

Eigenschaften externer Schlüssel werden im Abschnitt Externer Schlüssel der Registerkarte Kryptografische Konfiguration und im -`XksKeyConfiguration`Element der DescribeKey Antwort angezeigt.

Der Abschnitt External key (Externer Schlüssel) wird in der AWS KMS-Konsole nur für KMS-Schlüssel in externen Schlüsselspeichern angezeigt. Er enthält Informationen über den externen

Schlüssel, der dem KMS-Schlüssel zugeordnet ist. Der externe Schlüssel ist ein kryptografischer Schlüssel außerhalb von AWS, der als Schlüsselmaterial für den KMS-Schlüssel im externen Schlüsselspeicher dient. Wenn Sie mit dem KMS-Schlüssel verschlüsseln oder entschlüsseln, wird der Vorgang von Ihrem externen Schlüsselmanager unter Verwendung des angegebenen externen Schlüssels ausgeführt.

Die folgenden Werte werden im Abschnitt External key (Externer Schlüssel) angezeigt.

ID des externen Schlüssels

Die Kennung für den externen Schlüssel in seinem externen Schlüsselmanager. Diesen Wert verwendet der Proxy des externen Schlüsselspeichers, um den externen Schlüssel zu identifizieren. Sie geben die ID des externen Schlüssels an, wenn Sie den KMS-Schlüssel erstellen. Sie kann danach nicht mehr geändert werden. Wenn sich der Wert der ID des externen Schlüssels, den Sie zur Erstellung des KMS-Schlüssels verwendet haben, ändert oder ungültig wird, müssen Sie die Löschung des KMS-Schlüssel planen und einen neuen KMS-Schlüssel mit dem korrekten Wert der ID des externen Schlüssels erstellen.

Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher (Konsole)

Zeigen Sie die KMS-Schlüssel in einem externen Schlüsselspeicher (Konsole) wie folgt an:

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Um die AWS-Region zu ändern, verwenden Sie die Regionenauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Zur Identifizierung der KMS-Schlüssel in Ihrem externen Schlüsselspeicher fügen Sie Ihrer Schlüsseltabelle die Felder Origin (Ursprung) und Custom key store ID (Benutzerdefinierte Schlüsselspeicher-ID) hinzu. KMS-Schlüssel in einem externen Schlüsselspeicher haben den Ursprungswert External key store (Externer Schlüsselspeicher).

   Wählen Sie oben rechts das Zahnradsymbol und dann Origin (Ursprung) aus. Wählen Sie anschließend Custom key store (Benutzerdefinierte Schlüsselspeicher-ID) und dann Confirm (Bestätigen) aus.
5. Wählen Sie den Alias oder die Schlüssel-ID eines KMS-Schlüssels in einem externen Schlüsselspeicher aus.
6. Zum Anzeigen der spezifischen Eigenschaften von KMS-Schlüsseln in einem externen Schlüsselspeicher wählen Sie die Registerkarte Cryptographic configuration (Kryptografische

Konfiguration) aus. Spezielle Werte für KMS-Schlüssel in einem externen Schlüsselspeicher werden in den Abschnitten Custom key store (Benutzerdefinierter Schlüsselspeicher) und External key (Externer Schlüsselspeicher) angezeigt.

Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher (AWS KMS-API)

Zeigen Sie die KMS-Schlüssel in einem externen Schlüsselspeicher (-API) wie folgt an:

Sie verwenden dieselben AWS KMS API-Operationen, um die KMS-Schlüssel in einem externen Schlüsselspeicher anzuzeigen, die Sie für jeden KMS-Schlüssel verwenden würden, einschließlich ListKeysDescribeKey, und GetKeyPolicy. Der folgende `describe-key`-Vorgang in der AWS CLI zeigt zum Beispiel die speziellen Felder für einen KMS-Schlüssel in einem externen Schlüsselspeicher an. Vor der Ausführung eines Befehls wie diesem müssen Sie die ID des Beispiel-KMS-Schlüssels durch einen gültigen Wert ersetzen.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "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"
    }
  }
}
```

Verwenden von KMS-Schlüsseln in einem externen Schlüsselspeicher

Nachdem Sie einen KMS-Schlüssel zur symmetrischen Verschlüsselung in einem externen Schlüsselspeicher erstellt haben, können Sie ihn für die folgenden kryptografischen Vorgänge verwenden:

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Die Operationen zur symmetrischen Verschlüsselung, die asymmetrische Datenschlüsselpaare generieren, GenerateDataKeyPair und GenerateDataKeyPairWithoutPlaintext, werden in benutzerdefinierten Schlüsselspeichern nicht unterstützt.

Ein Verschlüsselungskontext wird für alle kryptografischen Vorgänge mit KMS-Schlüsseln in einem externen Schlüsselspeicher unterstützt. Wie immer ist die Verwendung eines Verschlüsselungskontexts eine bewährte Methode, die AWS KMS empfiehlt.

Wenn Sie Ihren KMS-Schlüssel in einer Anforderung verwenden, identifizieren Sie den KMS-Schlüssel durch Schlüssel-ID, Schlüssel-ARN, Alias oder Alias-ARN. Sie müssen den externen Schlüsselspeicher nicht angeben. Die Antwort enthält die gleichen Felder, die auch für alle anderen KMS-Schlüssel mit symmetrischer Verschlüsselung zurückgegeben werden. Wenn Sie jedoch einen KMS-Schlüssel in einem externen Schlüsselspeicher verwenden, werden die Ver- und Entschlüsselungsvorgänge von Ihrem externen Schlüsselmanager unter Verwendung des externen Schlüssels ausgeführt, der dem KMS-Schlüssel zugeordnet ist.

Um sicherzustellen, dass ein mit einem KMS-Schlüssel in einem externen Schlüsselspeicher verschlüsselter Geheimtext mindestens so sicher ist wie ein mit einem Standard-KMS-Schlüssel verschlüsselter Geheimtext, verwendet AWS KMS eine doppelte Verschlüsselung. Die Daten werden zunächst in AWS KMS mit AWS KMS-Schlüsselmaterial verschlüsselt. Dann werden sie von Ihrem externen Schlüsselmanager mit dem externen Schlüssel für den KMS-Schlüssel verschlüsselt. Um doppelt verschlüsselten Geheimtext zu entschlüsseln, wird der Geheimtext zunächst von Ihrem externen Schlüsselmanager mit dem externen Schlüssel für den KMS-Schlüssel entschlüsselt. Dann wird er in AWS KMS entschlüsselt, wobei das AWS KMS-Schlüsselmaterial für den KMS-Schlüssel verwendet wird.

Damit dies möglich ist, sind die folgenden Bedingungen erforderlich.

- Der Schlüsselstatus des KMS-Schlüssels muss `Enabled` lauten. Den Schlüsselstatus finden Sie im Feld Status für vom Kunden verwaltete Schlüssel in der AWS KMS Konsole oder im `KeyState` Feld in der DescribeKey Antwort.
- Der externe Schlüsselspeicher, der den KMS-Schlüssel hostet, muss mit seinem externen Schlüsselspeicher-Proxy verbunden sein, d. h. der Verbindungsstatus des externen Schlüsselspeichers muss `CONNECTED` sein.

  Sie können den Verbindungsstatus auf der Seite Externe Schlüsselspeicher in der AWS KMS Konsole oder in der DescribeCustomKeyStores Antwort anzeigen. Der Verbindungsstatus des externen Schlüsselspeicher wird auch auf der Detailseite für den KMS-Schlüssel in der AWS KMS-Konsole angezeigt. Wählen Sie auf der Detailseite die Registerkarte Cryptographic configuration (Kryptografische Konfiguration) und sehen Sie im Feld Connection state (Verbindungsstatus) im Abschnitt Custom key store (Benutzerdefinierter Schlüsselspeicher) nach.

  Wenn der Verbindungsstatus `DISCONNECTED` ist, müssen Sie zuerst eine Verbindung herstellen. Wenn der Verbindungsstatus `FAILED` lautet, müssen Sie das Problem lösen, den externen Schlüsselspeicher trennen und ihn dann verbinden. Entsprechende Anweisungen finden Sie unter Herstellen und Trennen der Verbindung eines externen Schlüsselspeichers.
- Der Proxy des externen Schlüsselspeichers muss in der Lage sein, den externen Schlüssel zu finden.
- Der externe Schlüssel muss aktiviert sein und er muss die Verschlüsselung und Entschlüsselung durchführen.

  Der Status des externen Schlüssels ist unabhängig von Änderungen des Schlüsselstatus des KMS-Schlüssels, einschließlich der Aktivierung und Deaktivierung des KMS-Schlüssels, und wird von diesen nicht beeinflusst. Ebenso ändert das Deaktivieren oder Löschen des externen Schlüssels nicht den Schlüsselstatus des KMS-Schlüssels, aber kryptografische Vorgänge, die den zugehörigen KMS-Schlüssel verwenden, schlagen fehl.

Wenn diese Bedingungen nicht erfüllt sind, schlägt die kryptografische Produktion fehl und AWS KMS gibt die Ausnahme `KMSInvalidStateException` zurück. Möglicherweise müssen Sie den externen Schlüsselspeicher erneut verbinden oder Tools Ihres externen Schlüsselmanagers verwenden, um Ihren externen Schlüssel neu zu konfigurieren oder zu reparieren. Weitere Informationen finden Sie unter the section called “Fehlerbehebung bei externen Schlüsselspeichern”.

Bei der Verwendung von KMS-Schlüsseln in einem externen Schlüsselspeicher ist zu beachten, dass die KMS-Schlüssel in jedem externen Schlüsselspeicher gemeinsam ein Anforderungskontingent für benutzerdefinierte Schlüsselspeicher für kryptografische Vorgänge nutzen. Wenn Sie das Kontingent überschreiten, gibt AWS KMS `ThrottlingException` zurück. Details zum Anforderungskontingent für benutzerdefinierte Schlüsselspeicher finden Sie unter Anforderungskontingente für benutzerdefinierte Schlüsselspeicher.

Planen der Löschung von KMS-Schlüsseln aus einem externen Schlüsselspeicher

Wenn Sie sich sicher sind, dass Sie einen AWS KMS key nicht mehr für kryptografische Operationen benötigen, können Sie die Löschung des KMS-Schlüssels planen. Gehen Sie dazu genau so vor wie beim Planen der Löschung anderer KMS-Schlüssel aus AWS KMS. Das Löschen eines KMS-Schlüssels aus einem externen Schlüsselspeicher hat keine Auswirkungen auf den externen Schlüssel, der als sein Schlüsselmaterial diente.

Sie können die geplante Löschung eines KMS-Schlüssels während der vorgegebenen Wartezeit abbrechen. Ein gelöschter KMS-Schlüssel kann jedoch nicht wiederhergestellt werden. Sie können einen symmetrischen KMS-Schlüssel in einem externen Schlüsselspeicher nicht wiederherstellen, selbst wenn Sie denselben externen Schlüssel verwenden. Da jeder symmetrische KMS-Schlüssel in einem externen Schlüsselspeicher über eindeutiges AWS KMS-Schlüsselmaterial und Metadaten verfügt, kann nur der AWS KMS-Schlüssel, der einen symmetrischen Geheimtext verschlüsselt hat, diesen entschlüsseln.

> **Warning**
>
> Das Löschen eines KMS-Schlüssels ist ein potentiell gefährliches Verfahren, wodurch alle Daten, die mit dem KMS-Schlüssel verschlüsselt wurden, nicht wieder entschlüsselt werden können. Bevor Sie das Löschen des KMS-Schlüssels planen, untersuchen Sie die frühere Nutzung des KMS-Schlüssels und erstellen Sie einen Amazon- CloudWatch Alarm, der Sie benachrichtigt, wenn jemand versucht, den KMS-Schlüssel zu verwenden, während er gelöscht werden soll. Es wird empfohlen, im Zweifelsfall den KMS-Schlüssel zu deaktivieren anstelle ihn zu löschen.

Wenn Sie die Löschung eines KMS-Schlüssels aus einem externen Schlüsselspeicher planen, ändert sich der Status des Schlüssels in Pending deletion (Löschung ausstehend). Der KMS-Schlüssel verbleibt auch während der Wartezeit im Status Pending deletion (Löschung ausstehend), selbst wenn der KMS-Schlüssel nicht mehr verfügbar ist, weil Sie die Verbindung zum externen

Schlüsselspeicher getrennt haben. Dies erlaubt es Ihnen, die Löschung des KMS-Schlüssels vor Ablauf der Wartezeit jederzeit abzubrechen. Nach Ablauf der Wartezeit löscht AWS KMS den KMS-Schlüssel aus AWS KMS.

Wenn Sie die Löschung eines KMS-Schlüssels aus einem externen Schlüsselspeicher planen, wird der KMS-Schlüssel sofort unbrauchbar (vorbehaltlich einer letztendlichen Konsistenz). Ressourcen, die mit durch den KMS-Schlüssel geschützten Datenschlüsseln verschlüsselt wurden, sind jedoch nicht betroffen, bis der KMS-Schlüssel erneut verwendet wird, z. B. zur Entschlüsselung des Datenschlüssels. Dieses Problem betrifft AWS-Services, von denen viele Datenschlüssel verwenden, um Ihre Ressourcen zu schützen. Details hierzu finden Sie unter Auswirkung von unbrauchbaren KMS-Schlüsseln auf Datenschlüssel.

Sie können die Planung, Stornierung und Löschung des KMS-Schlüssels in Ihren AWS CloudTrail-Protokollen überwachen.

Fehlerbehebung bei externen Schlüsselspeichern

Die Lösung der meisten Probleme mit externen Schlüsselspeichern wird durch die Fehlermeldung angezeigt, die bei jeder Ausnahme AWS KMS angezeigt wird, oder durch den Verbindungsfehlercode, der AWS KMS zurückgegeben wird, wenn ein Versuch, den externen Schlüsselspeicher mit seinem externen Schlüsselspeicher-Proxy zu verbinden, fehlschlägt. Einige Probleme sind jedoch etwas komplexer.

Wenn Sie ein Problem mit einem externen Schlüsselspeicher diagnostizieren, ermitteln Sie zunächst die Ursache. Dadurch wird die Bandbreite der potenziellen Abhilfemaßnahmen eingeschränkt und Ihre Fehlersuche wird effizienter.

- AWS KMS — Das Problem liegt möglicherweise darin AWS KMS, z. B. ein falscher Wert in der Konfiguration Ihres externen Schlüsselspeichers.
- Extern — Das Problem kann außerhalb von liegen AWS KMS, einschließlich Problemen mit der Konfiguration oder dem Betrieb des externen Schlüsselspeicher-Proxys, des externen Schlüsselmanagers, der externen Schlüssel oder des VPC-Endpunktdienstes.
- Netzwerk – Es könnte sich um ein Problem mit der Konnektivität oder dem Netzwerk handeln, z. B. um ein Problem mit Ihrem Proxy-Endpunkt, dem Port oder Ihrem privaten DNS-Namen oder Ihrer Domain.

> **Note**
>
> Wenn Verwaltungsoperationen für externe Schlüsselspeicher fehlschlagen, generieren sie verschiedene Ausnahmen. AWS KMS Kryptografische Operationen kehren jedoch `KMSInvalidStateException` bei allen Fehlern zurück, die mit der externen Konfiguration oder dem Verbindungsstatus des externen Schlüsselspeichers zusammenhängen. Identifizieren Sie das Problem anhand des zugehörigen Fehlermeldungstexts.
> Der ConnectCustomKeyStoreVorgang ist schnell erfolgreich, bevor der Verbindungsvorgang abgeschlossen ist. Um festzustellen, ob der Verbindungsvorgang erfolgreich ausgeführt wird, sehen Sie sich den Verbindungsstatus des externen Schlüsselspeichers an. Wenn der Verbindungsvorgang fehlschlägt, gibt AWS KMS einen Verbindungsfehlercode zurück, der Aufschluss über die Ursache gibt und Abhilfemaßnahmen vorschlägt.

Themen

Tools zur Fehlerbehebung bei externen Schlüsselspeichern

AWS KMS stellt mehrere Tools bereit, mit denen Sie Probleme mit Ihrem externen Schlüsselspeicher und seinen Schlüsseln identifizieren und lösen können. Verwenden Sie diese Tools zusammen mit den Tools, die mit Ihrem externen Schlüsselspeicher-Proxy und Ihrem externen Schlüsselmanager bereitgestellt werden.

> **Note**
>
> Ihr externer Schlüsselspeicher-Proxy und Ihr externer Schlüsselmanager bieten möglicherweise einfachere Methoden zum Erstellen und Verwalten Ihres externen Schlüsselspeichers und seiner KMS-Schlüssel. Weitere Informationen finden Sie in der Dokumentation Ihrer externen Tools.

AWS KMS Ausnahmen und Fehlermeldungen

AWS KMS bietet eine detaillierte Fehlermeldung zu allen auftretenden Problemen. Weitere Informationen zu AWS KMS Ausnahmen finden Sie in der AWS Key Management Service API-Referenz und in den AWS SDKs. Auch wenn Sie die AWS KMS Konsole verwenden, könnten diese Verweise für Sie hilfreich sein. Sehen Sie sich zum Beispiel die Fehlerliste für die `CreateCustomKeyStores`-Operation an.

Wenn das Problem in einem anderen AWS Dienst auftritt, z. B. wenn Sie einen KMS-Schlüssel in Ihrem externen Schlüsselspeicher verwenden, um eine Ressource in einem anderen AWS Dienst zu schützen, stellt der AWS Dienst möglicherweise zusätzliche Informationen zur Verfügung, mit denen Sie das Problem identifizieren können. Wenn der AWS Dienst die Meldung nicht bereitstellt, können Sie die Fehlermeldung in den CloudTrail Protokollen einsehen, in denen die Verwendung Ihres KMS-Schlüssels aufgezeichnet wird.

CloudTrail Logs

Jeder AWS KMS API-Vorgang, einschließlich Aktionen in der AWS KMS Konsole, wird in AWS CloudTrail Protokollen aufgezeichnet. AWS KMS zeichnet einen Protokolleintrag für erfolgreiche und fehlgeschlagene Operationen auf. Bei fehlgeschlagenen Operationen enthält der Protokolleintrag den Namen der AWS KMS -Ausnahme (`errorCode`) und die Fehlermeldung (`errorMessage`). Sie können diese Informationen verwenden, um den Fehler zu identifizieren und zu beheben. Ein Beispiel finden Sie unter Fehler bei der Entschlüsselung mit einem KMS-Schlüssel in einem externen Schlüsselspeicher.

Der Protokolleintrag enthält auch die Anfrage-ID. Wenn die Anfrage Ihren externen Schlüsselspeicher-Proxy erreicht hat, können Sie mithilfe der Anfrage-ID im Protokolleintrag die entsprechende Anfrage in Ihren Proxy-Protokollen finden, sofern Ihr Proxy sie bereitstellt.

CloudWatch Metriken

AWS KMS zeichnet detaillierte CloudWatch Amazon-Metriken über den Betrieb und die Leistung Ihres externen Schlüsselspeichers auf, darunter Latenz, Drosselung, Proxyfehler, den Status

des externen Schlüsselmanagers, die Anzahl der Tage bis zum Ablauf Ihres TLS-Zertifikats und das gemeldete Alter Ihrer Proxy-Authentifizierungsdaten. Sie können diese Metriken verwenden, um Datenmodelle für den Betrieb Ihres externen Schlüsselspeichers und CloudWatch Alarme zu entwickeln, die Sie vor drohenden Problemen warnen, bevor sie auftreten.

> **Important**
>
> AWS KMS empfiehlt, dass Sie CloudWatch Alarme erstellen, um die Messwerte des externen Schlüsselspeichers zu überwachen. Diese Alarme warnen Sie vor ersten Anzeichen von Problemen, bevor sie auftreten.

Überwachungsdiagramme

AWS KMS zeigt auf der Detailseite für jeden externen Schlüsselspeicher in der AWS KMS Konsole Diagramme der CloudWatch Messwerte für externe Schlüsselspeicher an. Sie können die Daten in den Diagrammen verwenden, um die Fehlerquelle zu lokalisieren, drohende Probleme zu erkennen, Ausgangswerte festzulegen und Ihre CloudWatch Alarmschwellenwerte zu verfeinern. Einzelheiten zur Interpretation der Überwachungsdiagramme und zur Verwendung der darin enthaltenen Daten finden Sie unter Überwachung eines externen Schlüsselspeichers.

Anzeigen von externen Schlüsselspeichern und KMS-Schlüsseln

AWS KMS zeigt detaillierte Informationen zu Ihren externen Schlüsselspeichern und den KMS-Schlüsseln im externen Schlüsselspeicher in der AWS KMS Konsole sowie in der Antwort auf die AND-Operationen an. DescribeCustomKeyStoresDescribeKey Diese Anzeigen enthalten spezielle Felder für externe Schlüsselspeicher und KMS-Schlüssel mit Informationen, die Sie für die Problembehandlung verwenden können, z. B. den Verbindungsstatus des externen Schlüsselspeichers und die ID des externen Schlüssels, der dem KMS-Schlüssel zugeordnet ist. Details dazu finden Sie unter Anzeigen eines externen Schlüsselspeichers und Anzeigen von KMS-Schlüsseln in einem externen Schlüsselspeicher.

XKS-Proxy-Testclient

AWS KMS stellt einen Open-Source-Testclient bereit, der überprüft, ob Ihr externer Schlüsselspeicher-Proxy der API-Spezifikation für den AWS KMS externen Schlüsselspeicher-Proxy entspricht. Sie können diesen Testclient verwenden, um Probleme mit Ihrem externen Schlüsselspeicher-Proxy zu identifizieren und zu beheben.

Konfigurationsfehler

Wenn Sie einen externen Schlüsselspeicher erstellen, geben Sie Eigenschaftswerte an, die die Konfiguration Ihres externen Schlüsselspeichers umfassen. Dazu zählen beispielsweise die Proxy-Authentifizierungsanmeldeinformation, der Proxy-URI-Endpunkt, der Proxy-URI-Pfad und der Name des VPC-Endpunktservice. Wenn ein Fehler in einem Eigenschaftswert AWS KMS erkannt wird, schlägt der Vorgang fehl und es wird ein Fehler zurückgegeben, der auf den fehlerhaften Wert hinweist.

Viele Konfigurationsprobleme lassen sich beheben, indem der falsche Wert korrigiert wird. Sie können einen ungültigen Proxy-URI-Pfad oder eine ungültige Proxy-Authentifizierungsanmeldeinformation korrigieren, ohne den externen Schlüsselspeicher zu trennen. Definitionen dieser Werte, einschließlich der Anforderungen hinsichtlich der Eindeutigkeit, finden Sie unter Erfüllen der Voraussetzungen. Anweisungen zum Aktualisieren dieser Werte finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.

Laden Sie beim Erstellen oder Aktualisieren Ihres externen Schlüsselspeichers eine Proxy-Konfigurationsdatei in die AWS KMS -Konsole hoch, um Fehler bei den Werten für den Proxy-URI-Pfad und die Proxy-Authentifizierungsanmeldeinformation zu vermeiden. Dies ist eine JSON-basierte Datei mit den Werten für den Proxy-URI-Pfad und die Proxy-Authentifizierungsanmeldeinformation, die von Ihrem externen Schlüsselspeicher-Proxy oder vom externen Schlüsselmanager bereitgestellt werden. Sie können keine Proxy-Konfigurationsdatei für AWS KMS API-Operationen verwenden, aber Sie können die Werte in der Datei verwenden, um Parameterwerte für Ihre API-Anfragen bereitzustellen, die den Werten in Ihrem Proxy entsprechen.

Allgemeine Konfigurationsfehler

Ausnahmen: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (kryptografische Operationen), `XksProxyInvalidConfigurationException` (Verwaltungsoperationen, außer `CreateKey`)

Verbindungsfehlercodes: `XKS_PROXY_INVALID_CONFIGURATION`, `XKS_PROXY_INVALID_TLS_CONFIGURATION`

AWS KMS Testet bei externen Schlüsselspeichern mit öffentlicher Endpunktkonnektivität die Eigenschaftswerte, wenn Sie den externen Schlüsselspeicher erstellen und aktualisieren. Für externe Schlüsselspeicher mit der Konnektivität eines VPC-Endpunktservice testet AWS KMS die Eigenschaftswerte, wenn Sie den externen Schlüsselspeicher verbinden und aktualisieren.

> **Note**
>
> Die `ConnectCustomKeyStore`-Operation ist asynchron und kann auch dann erfolgreich ausgeführt werden, wenn sich der externe Schlüsselspeicher nicht mit seinem externen Schlüsselspeicher-Proxy verbinden lässt. In diesem Fall gibt es keine Ausnahme, aber der Verbindungsstatus des externen Schlüsselspeichers ist fehlgeschlagen und ein Verbindungsfehlercode erklärt die Fehlermeldung. Weitere Informationen finden Sie unter Fehler bei der Verbindung mit dem externen Schlüsselspeicher.

Wenn ein Fehler in einem Eigenschaftswert AWS KMS erkannt wird, schlägt der Vorgang fehl `XksProxyInvalidConfigurationException` und es wird eine der folgenden Fehlermeldungen angezeigt.

Der externe Schlüsselspeicher-Proxy hat die Anfrage aufgrund eines ungültigen URI-Pfads abgelehnt. Überprüfen Sie den URI-Pfad für Ihren externen Schlüsselspeicher und aktualisieren Sie ihn gegebenenfalls.

- Der Proxy-URI-Pfad ist der Basispfad für AWS KMS Anfragen an die Proxy-APIs. Wenn dieser Pfad falsch ist, schlagen alle Anfragen an den Proxy fehl. Verwenden Sie die AWS KMS -Konsole oder die `DescribeCustomKeyStores`-Operation, um den aktuellen Proxy-URI-Pfad für Ihren externen Schlüsselspeicher anzuzeigen. Wie Sie den richtigen Proxy-URI-Pfad finden, erfahren Sie in der Dokumentation für Ihren externen Schlüsselspeicher-Proxy. Informationen zum Korrigieren des Werts für den Proxy-URI-Pfad finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.
- Der Proxy-URI-Pfad für Ihren externen Schlüsselspeicher-Proxy kann sich ändern, wenn Ihr externer Schlüsselspeicher-Proxy oder der externe Schlüsselmanager aktualisiert wird. Informationen zu diesen Änderungen finden Sie in der Dokumentation für Ihren externen Schlüsselspeicher-Proxy oder externen Schlüsselmanager.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`
AWS KMS kann keine TLS-Verbindung zum externen Schlüsselspeicher-Proxy herstellen. Überprüfen Sie die TLS-Konfiguration, einschließlich des Zertifikats.

- Für alle externen Schlüsselspeicher-Proxys ist ein TLS-Zertifikat erforderlich. Das TLS-Zertifikat muss von einer öffentlichen Zertifizierungsstelle (CA) ausgestellt werden, die für externe Schlüsselspeicher unterstützt wird. Eine Liste der unterstützten Zertifizierungsstellen finden Sie unter Vertrauenswürdige Zertifizierungsstellen in der API-Spezifikation von AWS KMS für externe Schlüsselspeicher-Proxys.
- Für die Konnektivität eines öffentlichen Endpunkts muss der Subject Common Name (CN) auf dem TLS-Zertifikat mit dem Domainnamen im Proxy-URI-Endpunkt für den externen Schlüsselspeicher-Proxy identisch sein. Ist der öffentliche Endpunkt beispielsweise https://myproxy.xks.example.com, muss der CN auf dem TLS-Zertifikat `myproxy.xks.example.com` oder `*.xks.example.com` lauten.
- Für die Konnektivität eines VPC-Endpunktservice muss der Subject Common Name (CN) auf dem TLS-Zertifikat mit dem privaten DNS-Namen für Ihren VPC-Endpunktservice übereinstimmen. Wenn der private DNS-Name beispielsweise myproxy-private.xks.example.com ist, muss der CN auf dem TLS-Zertifikat `myproxy-private.xks.example.com` oder `*.xks.example.com` lauten.
- Das TLS-Zertifikat darf nicht abgelaufen sein. Verwenden Sie SSL-Tools wie OpenSSL, um das Ablaufdatum eines TLS-Zertifikats zu ermitteln. Verwenden Sie die XksProxyCertificateDaysToExpire CloudWatch Metrik, um das Ablaufdatum eines TLS-Zertifikats zu überwachen, das einem externen Schlüsselspeicher zugeordnet ist. Die Anzahl der Tage bis zum Ablaufdatum Ihrer TLS-Zertifizierung wird auch im Bereich Überwachung der AWS KMS Konsole angezeigt.
- Wenn Sie die Konnektivität eines öffentlichen Endpunkts verwenden, testen Sie Ihre SSL-Konfiguration mithilfe von SSL-Testtools. TLS-Verbindungsfehler können durch eine falsche Zertifikatverkettung verursacht werden.

Konfigurationsfehler bei der Konnektivität eines VPC-Endpunktservice

Ausnahmen: `XksProxyVpcEndpointServiceNotFoundException`, `XksProxyVpcEndpointServiceInvalidConfigurationException`

Zusätzlich zu allgemeinen Verbindungsproblemen können beim Erstellen, Verbinden oder Aktualisieren eines externen Schlüsselspeichers mit VPC-Endpunktdienst-Konnektivität die folgenden Probleme auftreten. AWS KMS testet die Eigenschaftswerte eines externen Schlüsselspeichers mit VPC-Endpunktdienst-Konnektivität, während der externe Schlüsselspeicher erstellt, verbunden und aktualisiert wird. Wenn Verwaltungsoperationen aufgrund von Konfigurationsfehlern fehlschlagen, werden die folgenden Ausnahmen generiert:

XksProxyVpcEndpointServiceNotFoundException

Dies kann folgende Ursachen haben:

- Ein falscher Name des VPC-Endpunktservice. Stellen Sie sicher, dass der Name des VPC-Endpunktservice für den externen Schlüsselspeicher korrekt ist und mit dem Wert des Proxy-URI-Endpunkts für den externen Schlüsselspeicher übereinstimmt. Um den VPC-Endpunkt-Servicenamen zu finden, verwenden Sie die Amazon VPC-Konsole oder den DescribeVpcEndpointServicesVorgang. Verwenden Sie die AWS KMS Konsole oder den DescribeCustomKeyStoresVorgang, um den VPC-Endpunktdienstnamen und den Proxy-URI-Endpunkt eines vorhandenen externen Schlüsselspeichers zu ermitteln. Details hierzu finden Sie unter Anzeigen eines externen Schlüsselspeichers.
- Der VPC-Endpunktdienst befindet sich möglicherweise in einem anderen AWS-Region als dem externen Schlüsselspeicher. Stellen Sie sicher, dass sich der VPC-Endpunktservice und der externe Schlüsselspeicher in derselben Region befinden. (Der externe Name des Regionsnamens, z. B.`us-east-1`, ist Teil des VPC-Endpunktdienstnamens, z. B. com.amazonaws.vpce.us-east-1.vpce-svc-example.) Eine Liste der Anforderungen für den VPC-Endpunktservice für einen externen Schlüsselspeicher finden Sie unter VPC-Endpunktservice. Ein VPC-Endpunktservice oder ein externer Schlüsselspeicher lässt sich nicht in eine andere Region verschieben. Sie können jedoch einen neuen externen Schlüsselspeicher in derselben Region wie der VPC-Endpunktservice erstellen. Details dazu finden Sie unter Konfigurieren der Konnektivität eines VPC-Endpunktservice und Erstellen eines externen Schlüsselspeichers.
- AWS KMS ist kein zulässiger Principal für den VPC-Endpunktdienst. Die Liste der zulässigen Prinzipale für den VPC-Endpunktservice muss den `cks.kms.<region>.amazonaws.com`-Wert enthalten, etwa `cks.kms.eu-west-3.amazonaws.com`. Anweisungen zum Hinzufügen dieses Werts finden Sie unter Verwalten von Berechtigungen im AWS PrivateLink -Handbuch.

XksProxyVpcEndpointServiceInvalidConfigurationException

Dieser Fehler tritt auf, wenn der VPC-Endpunktservice eine der folgenden Anforderungen nicht erfüllt:

- Die VPC muss über mindestens zwei private Subnetze verfügen, die in verschiedenen Availability Zones sind. Weitere Informationen zum Hinzufügen eines Subnetzes zu Ihrer VPC finden Sie unter Erstellen eines Subnetzes in der VPC im Amazon VPC-Benutzerhandbuch.

- Der Typ es VPC-Endpunktservice muss einen Network Load Balancer verwenden, keinen Gateway Load Balancer.
- Für den VPC-Endpunktservice darf keine Akzeptanz erforderlich sein (Akzeptanz erforderlich muss auf falsch gesetzt sein.). Wenn jede Verbindungsanforderung manuell akzeptiert werden muss, AWS KMS kann der VPC-Endpunktdienst nicht verwendet werden, um eine Verbindung zum externen Schlüsselspeicher-Proxy herzustellen. Einzelheiten hierzu finden Sie unter Annehmen oder Ablehnen von Verbindungsanforderungen im AWS PrivateLink -Handbuch.
- Der VPC-Endpunktservice muss einen privaten DNS-Namen haben, der eine Subdomain einer öffentlichen Domain ist. Lautet der private DNS-Name beispielsweise `https://myproxy-private.xks.example.com`, müssen die Domains `xks.example.com` und `example.com` über einen öffentlichen DNS-Server verfügen. Informationen zum Anzeigen oder Ändern des privaten DNS-Namens für Ihren VPC-Endpunktservice finden Sie unter Verwalten von DNS-Namen für VPC-Endpunktservices im AWS PrivateLink -Handbuch.
- Der Domain-Verifizierungsstatus der Domain für Ihren privaten DNS-Namen muss `verified` lauten. Informationen zum Anzeigen und Aktualisieren des Verifizierungsstatus der Domain für Ihren privaten DNS-Namen finden Sie unter Verifizieren der Domain Ihres privaten DNS-Namens. Nachdem Sie den erforderlichen Textdatensatz hinzugefügt haben, kann es einige Minuten dauern, bis der aktualisierte Verifizierungsstatus angezeigt wird.

  > **Note**
  >
  > Eine private DNS-Domain kann nur verifiziert werden, wenn es sich um die Subdomain einer öffentlichen Domain handelt. Andernfalls ändert sich der Verifizierungsstatus der privaten DNS-Domain nicht, auch wenn Sie den erforderlichen TXT-Datensatz hinzugefügt haben.

- Der private DNS-Name des VPC-Endpunktservice muss mit dem Wert des Proxy-URI-Endpunkts für den externen Schlüsselspeicher übereinstimmen. Für einen externen Schlüsselspeicher mit der Konnektivität eines VPC-Endpunktservice muss der Proxy-URI-Endpunkt `https://` gefolgt vom privaten DNS-Namen des VPC-Endpunktservice sein. Informationen zum Anzeigen des Werts für den Proxy-URI-Endpunkt finden Sie unter Anzeigen eines externen Schlüsselspeichers. Informationen zum Ändern des Werts für den Proxy-URI-Endpunkt finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers.

Fehler bei der Verbindung mit dem externen Schlüsselspeicher

Das Verbinden eines externen Schlüsselspeichers mit seinem externen Schlüsselspeicher-Proxy dauert etwa fünf Minuten. Sofern sie nicht schnell fehlschlägt, gibt die `ConnectCustomKeyStore`-Operation eine HTTP-Antwort 200 und ein JSON-Objekt ohne Eigenschaften zurück. Diese erste Antwort gibt jedoch nicht an, dass die Verbindung erfolgreich war. Um festzustellen, ob der externe Schlüsselspeicher verbunden ist, sehen Sie sich den Verbindungsstatus an. Wenn die Verbindung fehlschlägt, ändert sich der Verbindungsstatus des externen Schlüsselspeichers in einen Verbindungsfehlercode, der die Ursache des Fehlers erklärt, `FAILED` und es wird ein Verbindungsfehlercode AWS KMS zurückgegeben.

> **Note**
>
> Wenn der Verbindungsstatus eines benutzerdefinierten Schlüsselspeichers `FAILED` ist, müssen Sie den benutzerdefinierten Schlüsselspeicher trennen, bevor Sie versuchen, ihn wieder zu verbinden. Ein benutzerdefinierter Schlüsselspeicher mit dem Verbindungsstatus `FAILED` kann nicht verbunden werden.

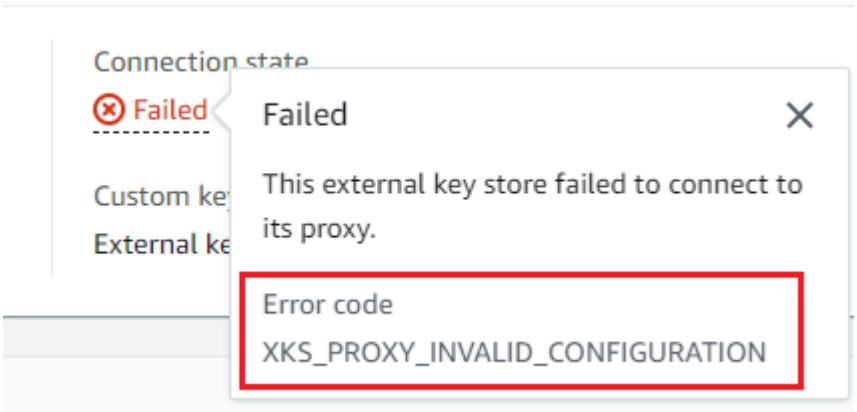
Anzeigen des Verbindungsstatus eines externen Schlüsselspeichers:

- Sehen Sie sich in der DescribeCustomKeyStoresAntwort den Wert des `ConnectionState` Elements an.
- In der AWS KMS Konsole wird der Verbindungsstatus in der Tabelle mit dem externen Schlüsselspeicher angezeigt. Außerdem wird auf der Detailseite für jeden externen Schlüsselspeicher der Verbindungsstatus im Abschnitt Allgemeine Konfiguration angezeigt.

Wenn der Verbindungsstatus `FAILED` lautet, lässt sich mithilfe des Verbindungsfehlercodes der Fehler erklären.

Anzeigen des Verbindungsfehlercodes:

- Sehen Sie sich in der DescribeCustomKeyStoresAntwort den Wert des `ConnectionErrorCode` Elements an. Die `DescribeCustomKeyStores`-Antwort enthält dieses Element nur, wenn der `ConnectionState` `FAILED` ist.
- Um den Verbindungsfehlercode in der AWS KMS Konsole auf der Detailseite für den externen Schlüsselspeicher anzuzeigen, zeigen Sie mit der Maus auf den Wert Fehlgeschlagen.



Verbindungsfehlercodes für externe Schlüsselspeicher

Die folgenden Verbindungsfehlercodes gelten für externe Schlüsselspeicher

`INTERNAL_ERROR`

AWS KMS konnte die Anfrage aufgrund eines internen Fehlers nicht abschließen. Wiederholen Sie die Anforderung. Trennen Sie bei `ConnectCustomKeyStore`-Anforderungen den benutzerdefinierten Schlüsselspeicher, bevor Sie die Verbindung wiederherstellen.

`INVALID_CREDENTIALS`

Einer oder beide `XksProxyAuthenticationCredential`-Werte sind auf dem angegebenen externen Schlüsselspeicher-Proxy ungültig.

`NETWORK_ERRORS`

Netzwerkfehler AWS KMS verhindern, dass der benutzerdefinierte Schlüsselspeicher mit seinem Backing-Schlüsselspeicher verbunden werden kann.

`XKS_PROXY_ACCESS_DENIED`

AWS KMS Anfragen wird der Zugriff auf den externen Schlüsselspeicher-Proxy verweigert. Wenn es für den externen Schlüsselspeicher-Proxy Autorisierungsregeln gibt, stellen Sie sicher, dass diese zulassen, dass AWS KMS in Ihrem Namen mit dem Proxy kommuniziert.

`XKS_PROXY_INVALID_CONFIGURATION`

Ein Konfigurationsfehler verhindert, dass der externe Schlüsselspeicher eine Verbindung zu seinem Proxy herstellt. Überprüfen Sie den Wert von `XksProxyUriPath`.

`XKS_PROXY_INVALID_RESPONSE`

AWS KMS kann die Antwort des externen Schlüsselspeicher-Proxys nicht interpretieren. Wenn Sie diesen Verbindungsfehlercode wiederholt sehen, benachrichtigen Sie den Proxy-Anbieter Ihres externen Schlüsselspeichers.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`

AWS KMS kann keine Verbindung zum externen Schlüsselspeicher-Proxy herstellen, da die TLS-Konfiguration ungültig ist. Stellen Sie sicher, dass der externe Schlüsselspeicher-Proxy TLS 1.2 oder 1.3 unterstützt. Stellen Sie außerdem sicher, dass das TLS-Zertifikat nicht abgelaufen ist, dass es mit dem Hostnamen im `XksProxyUriEndpoint`-Wert übereinstimmt und dass es von einer vertrauenswürdigen Zertifizierungsstelle signiert wurde, die in der Liste der vertrauenswürdigen Zertifizierungsstellen enthalten ist.

`XKS_PROXY_NOT_REACHABLE`

AWS KMS kann nicht mit Ihrem externen Schlüsselspeicher-Proxy kommunizieren. Stellen Sie sicher, dass der `XksProxyUriEndpoint` und der `XksProxyUriPath` korrekt sind. Überprüfen Sie mithilfe der Tools für Ihren externen Schlüsselspeicher-Proxy, ob der Proxy aktiv und in seinem Netzwerk verfügbar ist. Stellen Sie außerdem sicher, dass die Instances Ihres externen Schlüsselmanagers ordnungsgemäß funktionieren. Verbindungsversuche schlagen mit diesem Verbindungsfehlercode fehl, wenn der Proxy meldet, dass keine Instance von externen Schlüsselmanagern verfügbar ist.

`XKS_PROXY_TIMED_OUT`

AWS KMS kann eine Verbindung zum externen Schlüsselspeicher-Proxy herstellen, aber der Proxy reagiert nicht AWS KMS innerhalb der zugewiesenen Zeit. Wenn Sie diesen Verbindungsfehlercode wiederholt sehen, benachrichtigen Sie den Proxy-Anbieter Ihres externen Schlüsselspeichers.

`XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION`

Die Konfiguration des Amazon VPC-Endpunktdienstes entspricht nicht den Anforderungen für einen AWS KMS externen Schlüsselspeicher.

- Der VPC-Endpunktservice muss ein Endpunktservice für Schnittstellen-Endpunkte im AWS-Konto des Aufrufers sein.
- Er muss über einen Network Load Balancer (NLB) verfügen, der mit mindestens zwei Subnetzen verbunden ist, die jeweils in einer anderen Availability Zone liegen.

- Die `Allow principals` Liste muss den AWS KMS Service Principal für die Region enthaltencks.kms.<region>.amazonaws.com, z. B. `cks.kms.us-east-1.amazonaws.com`
- Er darf keine Annahme von Verbindungsanforderungen verlangen.
- Er muss einen privaten DNS-Namen haben. Der private DNS-Name für einen externen Schlüsselspeicher mit `VPC_ENDPOINT_SERVICE`-Konnektivität muss in seiner AWS-Region eindeutig sein.
- Der Verifizierungsstatus der Domain des privaten DNS-Namens muss `verified` lauten.
- Das TLS-Zertifikat gibt den privaten DNS-Hostnamen an, unter dem der Endpunkt erreichbar ist.

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

AWS KMS kann den VPC-Endpunktdienst, den er für die Kommunikation mit dem externen Schlüsselspeicher-Proxy verwendet, nicht finden. Stellen Sie sicher, dass der `XksProxyVpcEndpointServiceName` korrekt ist und der AWS KMS -Service-Prinzipal über Service-Verbraucher-Berechtigungen für den Amazon VPC-Endpunktservice verfügt.

Latenz- und Zeitüberschreitungsfehler

Ausnahmen: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (kryptografische Operationen), `XksProxyUriUnreachableException` (Verwaltungsoperationen)

Verbindungsfehlercodes: `XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Wenn der Proxy innerhalb des Timeout-Intervalls von 250 Millisekunden nicht kontaktiert werden AWS KMS kann, wird eine Ausnahme zurückgegeben. `CreateCustomKeyStore`und `UpdateCustomKeyStore` kehre zurück. `XksProxyUriUnreachableException` Kryptografische Operationen geben die Standard-`KMSInvalidStateException` mit einer Fehlermeldung zurück, die das Problem beschreibt. Falls dies `ConnectCustomKeyStore` fehlschlägt, wird ein Verbindungsfehlercode AWS KMS zurückgegeben, der das Problem beschreibt.

Zeitüberschreitungsfehler können vorübergehende Probleme sein, die sich durch das Wiederholen der Anforderung beheben lassen. Wenn dieses Problem weiterhin auftritt, überprüfen Sie, ob Ihr externer Schlüsselspeicher-Proxy aktiv und mit dem Netzwerk verbunden ist und ob sein Proxy-URI-Endpunkt, der Proxy-URI-Pfad und der Name des VPC-Endpunktservice (sofern vorhanden) in Ihrem externen Schlüsselspeicher korrekt sind. Stellen Sie außerdem sicher, dass sich Ihr externer

Schlüsselmanager in der Nähe des AWS-Region für Ihren externen Schlüsselspeicher befindet. Wenn Sie einen dieser Werte aktualisieren müssen, finden Sie unter Bearbeiten der Eigenschaften eines externen Schlüsselspeichers weitere Informationen.

Um Latenzmuster zu verfolgen, verwenden Sie die `XksProxyLatency` CloudWatch Metrik und das Diagramm mit der durchschnittlichen Latenz (basierend auf dieser Metrik) im Bereich Überwachung der AWS KMS Konsole. Ihr externer Schlüsselspeicher-Proxy generiert möglicherweise ebenfalls Protokolle und Metriken, die Latenz und Zeitüberschreitungen erfassen.

`XksProxyUriUnreachableException`
AWS KMS kann nicht mit dem externen Schlüsselspeicher-Proxy kommunizieren. Dies könnte ein vorübergehendes Netzwerkproblem sein. Wenn dieser Fehler wiederholt auftritt, überprüfen Sie, ob Ihr externer Schlüsselspeicher-Proxy aktiv und mit dem Netzwerk verbunden ist und ob sein Endpunkt-URI in Ihrem externen Schlüsselspeicher korrekt sind.

- Der externe Schlüsselspeicher-Proxy hat innerhalb des Timeout-Intervalls von 250 Millisekunden nicht auf eine AWS KMS Proxy-API-Anfrage geantwortet. Dies kann auf ein vorübergehendes Netzwerkproblem oder ein Betriebs- oder Leistungsproblem mit dem Proxy hinweisen. Wenn ein erneuter Versuch das Problem nicht löst, benachrichtigen Sie den Administrator für Ihren externen Schlüsselspeicher-Proxy.

Latenz- und Zeitüberschreitungsfehler äußern sich häufig als Verbindungsfehler. Wenn der ConnectCustomKeyStoreVorgang fehlschlägt, ändert sich der Verbindungsstatus des externen Schlüsselspeichers in einen Verbindungsfehlercode, der den Fehler `FAILED` erklärt, und es wird ein Verbindungsfehlercode AWS KMS zurückgegeben. Eine Liste der Verbindungsfehlercodes und Vorschläge zur Behebung der Fehler finden Sie unter Verbindungsfehlercodes für externe Schlüsselspeicher. Die Verbindungscodelisten für alle benutzerdefinierten Schlüsselspeicher und externe Schlüsselspeicher gelten für externe Schlüsselspeicher. Die folgenden Verbindungsfehler stehen im Zusammenhang mit der Latenz und Zeitüberschreitungen.

`XKS_PROXY_NOT_REACHABLE`

–oder–

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`

AWS KMS kann nicht mit dem externen Schlüsselspeicher-Proxy kommunizieren. Überprüfen Sie, ob Ihr externer Schlüsselspeicher-Proxy aktiv und mit dem Netzwerk verbunden ist und ob sein URI-Pfad und Endpunkt-URI oder der Name des VPC-Service in Ihrem externen Schlüssel speicher korrekt sind.

Dieser Fehler kann aus folgenden Gründen auftreten:

- Der externe Schlüsselspeicher-Proxy ist nicht aktiv bzw. nicht mit dem Netzwerk verbunden.
- Bei den Werten für den Proxy-URI-Endpunkt, den Proxy-URI-Pfad oder den Namen des VPC-Endpunktservice (sofern zutreffend) in der Konfiguration des externen Schlüsselspeichers ist ein Fehler aufgetreten. Um die Konfiguration des externen Schlüsselspeichers anzuzeigen, verwenden Sie den DescribeCustomKeyStoresVorgang oder rufen Sie die Detailseite für den externen Schlüsselspeicher in der AWS KMS Konsole auf.
- Möglicherweise liegt ein Netzwerkkonfigurationsfehler, z. B. ein Portfehler, auf dem Netzwerkpfad zwischen dem externen Schlüsselspeicher-Proxy AWS KMS und dem externen Schlüsselspeicher-Proxy vor. AWS KMS kommuniziert mit dem externen Schlüsselspeicher-Proxy auf Port 443. Dieser Wert kann nicht konfiguriert werden.
- Wenn der externe Schlüsselspeicher-Proxy (in einer GetHealthStatusAntwort) meldet, dass alle externen Schlüsselmanager-Instanzen vorhanden sind`UNAVAILABLE`, schlägt der ConnectCustomKeyStoreVorgang mit einem `ConnectionErrorCode` von `XKS_PROXY_NOT_REACHABLE` fehl. Weitere Informationen hierzu finden Sie in der Dokumentation Ihres externen Schlüsselmanagers.
- Dieser Fehler kann auf eine große physische Entfernung zwischen dem externen Schlüsselmanager und AWS-Region dem externen Schlüsselspeicher zurückzuführen sein. Die Ping-Latenz (Network Round-Trip Time (RTT)) zwischen dem AWS-Region und dem externen Schlüsselmanager darf nicht mehr als 35 Millisekunden betragen. Möglicherweise müssen Sie einen externen Schlüsselspeicher in einem Bereich einrichten AWS-Region , der sich näher am externen Schlüsselmanager befindet, oder den externen Schlüsselmanager in ein Rechenzentrum verschieben, das näher am. AWS-Region

`XKS_PROXY_TIMED_OUT`

–oder–

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
AWS KMS hat die Anforderung abgelehnt, da der externe Schlüsselspeicher-Proxy nicht rechtzeitig geantwortet hat. Wiederholen Sie die Anforderung. Wenn Sie diesen Fehler wiederholt sehen, melden Sie ihn dem Administrator für Ihren externen Schlüsselspeicher-Proxy.

Dieser Fehler kann aus folgenden Gründen auftreten:

- Dieser Fehler kann darauf zurückzuführen sein, dass der externe Schlüsselmanager und der externe Schlüsselspeicher-Proxy geografisch weit voneinander entfernt sind. Bringen Sie den externen Schlüsselspeicher-Proxy nach Möglichkeit näher an den externen Schlüsselmanager.
- Timeoutfehler können auftreten, wenn der Proxy nicht darauf ausgelegt ist, das Volumen und die Häufigkeit der Anfragen von AWS KMS zu verarbeiten. Wenn Ihre CloudWatch Messwerte auf ein anhaltendes Problem hinweisen, benachrichtigen Sie Ihren Proxyadministrator für den externen Schlüsselspeicher.
- Zeitüberschreitungsfehler können auftreten, wenn die Verbindung zwischen dem externen Schlüsselmanager und der Amazon VPC für den externen Schlüsselspeicher nicht ordnungsgemäß funktioniert. Wenn Sie verwenden AWS Direct Connect, stellen Sie sicher, dass Ihre VPC und der externe Schlüsselmanager effektiv kommunizieren können. Hilfe zur Lösung von Problemen finden Sie AWS Direct Connect im AWS Direct Connect Benutzerhandbuch unter Problembehandlung.

`XKS_PROXY_TIMED_OUT`

–oder–

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
Der externe Schlüsselspeicher-Proxy hat nicht in der vorgesehenen Zeit auf die Anforderung geantwortet. Wiederholen Sie die Anforderung. Wenn Sie diesen Fehler wiederholt sehen, melden Sie ihn dem Administrator für Ihren externen Schlüsselspeicher-Proxy.

- Dieser Fehler kann darauf zurückzuführen sein, dass der externe Schlüsselmanager und der externe Schlüsselspeicher-Proxy geografisch weit voneinander entfernt sind. Bringen Sie den externen Schlüsselspeicher-Proxy nach Möglichkeit näher an den externen Schlüsselmanager.

Fehler mit der Anmeldeinformation für die Authentifizierung

Ausnahmen: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (kryptografische Operationen), `XksProxyIncorrectAuthenticationCredentialException` (Verwaltungsoperationen außer `CreateKey`)

Sie richten Anmeldeinformationen für die Authentifizierung AWS KMS auf Ihrem externen Schlüsselspeicher-Proxy ein und verwalten diese. Anschließend geben Sie die Werte AWS KMS der Anmeldeinformationen an, wenn Sie einen externen Schlüsselspeicher erstellen. Wenn Sie die Anmeldeinformation für die Authentifizierung ändern möchten, nehmen Sie diese Änderung auf Ihrem externen Schlüsselspeicher-Proxy vor. Aktualisieren Sie danach die Anmeldeinformation für Ihren externen Schlüsselspeicher. Wenn Ihr Proxy die Anmeldeinformation rotiert, müssen Sie sie für Ihren externen Schlüsselspeicher aktualisieren.

Wenn der externe Schlüsselspeicher-Proxy eine mit der Proxy-Authentifizierungsanmeldeinformation für Ihren externen Schlüsselspeicher signierte Anforderung nicht authentifiziert, hängt es von der Anforderung ab, was geschieht:

- `CreateCustomKeyStore` und `UpdateCustomKeyStore` schlagen mit einem `XksProxyIncorrectAuthenticationCredentialException` fehl.
- `ConnectCustomKeyStore` wird erfolgreich ausgeführt, aber die Verbindung schlägt fehl. Der Verbindungsstatus ist `FAILED` und der Verbindungsfehlercode lautet `INVALID_CREDENTIALS`. Details hierzu finden Sie unter Fehler bei der Verbindung mit dem externen Schlüsselspeicher.
- Kryptografische Operationen geben `KMSInvalidStateException` für alle externen Konfigurationsfehler und Verbindungsstatusfehler in einem externen Schlüsselspeicher zurück. Die zugehörige Fehlermeldung beschreibt das Problem.

Der externe Schlüsselspeicher-Proxy hat die Anforderung abgelehnt, da er AWS KMS nicht authentifizieren konnte. Überprüfen Sie die Anmeldeinformationen für Ihren externen Schlüssel speicher und aktualisieren Sie sie gegebenenfalls.

Dieser Fehler kann aus folgenden Gründen auftreten:

- Die Zugriffsschlüssel-ID oder der geheime Zugriffsschlüssel für den externen Schlüsselspeicher stimmt nicht mit den Werten überein, die auf dem externen Schlüsselspeicher-Proxy festgelegt wurden.

  Um diesen Fehler zu beheben, aktualisieren Sie die Proxy-Authentifizierungsanmeldeinformation für Ihren externen Schlüsselspeicher. Sie können diese Änderung vornehmen, ohne Ihren externen Schlüsselspeicher zu trennen.
- Ein Reverse-Proxy zwischen AWS KMS und dem externen Schlüsselspeicher-Proxy könnte HTTP-Header so manipulieren, dass die SigV4-Signaturen ungültig werden. Um diesen Fehler zu beheben, benachrichtigen Sie den Administrator für Ihren Proxy.

Fehler mit dem Schlüsselstatus

Ausnahmen: `KMSInvalidStateException`

`KMSInvalidStateException` wird für zwei verschiedene Zwecke für KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern verwendet.

- Wenn ein Verwaltungsvorgang (z. B. `CancelKeyDeletion`) fehlschlägt und diese Ausnahme zurückgibt, bedeutet dies, dass der Schlüsselstatus des KMS-Schlüssels nicht mit der Operation kompatibel ist.
- Wenn eine kryptografische Operation für einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher mit `KMSInvalidStateException` fehlschlägt, kann dies auf ein Problem mit dem Schlüsselstatus des KMS-Schlüssels hindeuten. `KMSInvalidStateException`Bei allen externen Konfigurationsfehlern und Verbindungsstatusfehlern in einem externen Schlüsselspeicher wird jedoch ein AWS KMS kryptografischer Vorgang zurückgegeben. Identifizieren Sie das Problem anhand der Fehlermeldung, die zusammen mit der Ausnahme angezeigt wird.

Informationen zum erforderlichen Schlüsselstatus für AWS KMS API-Operationen finden Sie unterWichtige Zustände von AWS KMS Schlüsseln. Um den Schlüsselstatus eines KMS-Schlüssels zu ermitteln, zeigen Sie auf der Seite Customer managed keys (Kundenverwaltete Schlüssel) das Feld Status des KMS-Schlüssels an. Oder verwenden Sie die DescribeKeyOperation und sehen Sie sich das `KeyState` Element in der Antwort an. Details hierzu finden Sie unter Anzeigen von Schlüsseln.

> **Note**
>
> Der Schlüsselstatus eines KMS-Schlüssels in einem externen Schlüsselspeicher sagt nichts über den Status des zugehörigen externen Schlüssels aus. Informationen zum Status des externen Schlüssels finden Sie mithilfe Ihres externen Schlüsselmanagers und der Tools für den externen Schlüsselspeicher-Proxy.
> Die `CustomKeyStoreInvalidStateException` bezieht sich auf den Verbindungsstatus des externen Schlüsselspeichers, nicht auf den Schlüsselstatus eines KMS-Schlüssels.

Eine kryptografische Operation für einen KMS-Schlüssel in einem benutzerdefinierten Speicher schlägt möglicherweise fehl, weil der Schlüsselstatus des KMS-Schlüssels `Unavailable` oder `PendingDeletion` lautet. (Deaktivierte Schlüssel geben `DisabledException` zurück.)

- Ein KMS-Schlüssel hat nur dann einen `Disabled` Schlüsselstatus, wenn Sie den KMS-Schlüssel in der AWS KMS Konsole oder mithilfe des DisableKeyVorgangs absichtlich deaktivieren. Wenn ein KMS-Schlüssel deaktiviert ist, können Sie ihn anzeigen und verwalten, ihn jedoch nicht für kryptographische Operationen verwenden. Um dieses Problem zu beheben, aktivieren Sie den Schlüssel. Details hierzu finden Sie unter Aktivieren und Deaktivieren von Schlüsseln.
- Ein KMS-Schlüssel hat den Schlüsselstatus `Unavailable`, wenn der externe Schlüsselspeicher von seinem externen Schlüsselspeicher-Proxy getrennt ist. Um die Nichtverfügbarkeit eines KMS-Schlüssels zu beheben, verbinden Sie den externen Schlüsselspeicher wieder. Wenn der externe Schlüsselspeicher wieder verbunden ist, wechselt der Schlüsselstatus der KMS-Schlüssel im externen Schlüsselspeicher automatisch wieder zum vorherigen Status, etwa zu `Enabled` oder `Disabled`.

  Ein KMS-Schlüssel hat den Schlüsselstatus `PendingDeletion`, wenn seine Löschung geplant ist und er darauf wartet. Ein Schlüsselstatusfehler bei einem KMS-Schlüssel, dessen Löschung aussteht, bedeutet, dass der Schlüssel nicht gelöscht werden sollte – entweder, weil er für die Verschlüsselung verwendet wird oder weil er für die Entschlüsselung erforderlich ist. Um den KMS-Schlüssel erneut zu aktivieren, brechen Sie den geplanten Löschvorgang ab und aktivieren Sie dann den Schlüssel. Details hierzu finden Sie unter Planen und Abbrechen der Löschung eines Schlüssels.

Entschlüsselungsfehler

Ausnahmen: `KMSInvalidStateException`

Wenn ein Entschlüsselungsvorgang mit einem KMS-Schlüssel in einem externen Schlüsselspeicher fehlschlägt, wird der Standard AWS KMS `KMSInvalidStateException` zurückgegeben, den kryptografische Operationen für alle externen Konfigurationsfehler und Verbindungsstatusfehler in einem externen Schlüsselspeicher verwenden. Die Fehlermeldung, die das Problem angibt.

Zum Entschlüsseln eines Geheimtexts, der mit doppelter Verschlüsselung verschlüsselt wurde, verwendet der externe Schlüsselmanager zunächst den externen Schlüssel, um die äußere Geheimtextschicht zu entschlüsseln. AWS KMS Verwendet dann das AWS KMS Schlüsselmaterial im KMS-Schlüssel, um die innere Chiffretextschicht zu entschlüsseln. Ein ungültiger oder beschädigter Geheimtext kann vom externen Schlüsselmanager oder von AWS KMS abgelehnt werden.

Die folgenden Fehlermeldungen werden zusammen mit der `KMSInvalidStateException` angezeigt, wenn die Entschlüsselung fehlschlägt. Sie deuten auf ein Problem mit dem Geheimtext oder dem optionalen Verschlüsselungskontext in der Anfrage hin.

Der externe Schlüsselspeicher-Proxy hat die Anforderung, da der angegebene Geheimtext oder zusätzliche authentifizierte Daten beschädigt sind, fehlen oder anderweitig ungültig sind.

- Wenn der externe Schlüsselspeicher-Proxy oder der externe Schlüsselmanager meldet, dass ein Chiffretext oder sein Verschlüsselungskontext ungültig ist, deutet dies in der Regel auf ein Problem mit dem Chiffretext oder dem Verschlüsselungskontext in der Anforderung hin, an die gesendet wurde. `Decrypt` AWS KMS Bei `Decrypt` Vorgängen AWS KMS sendet der Proxy denselben Chiffretext und denselben Verschlüsselungskontext, den er in der Anfrage empfängt. `Decrypt`

  Dieser Fehler kann durch ein Netzwerkproblem während der Übertragung verursacht werden, etwa durch ein „umgekipptes“ Bit. Wiederholen Sie die `Decrypt`-Anforderung. Wenn das Problem weiterhin besteht, stellen Sie sicher, dass der Geheimtext nicht geändert oder beschädigt wurde. Stellen Sie außerdem sicher, dass der Verschlüsselungskontext in der `Decrypt` Anfrage mit dem Verschlüsselungskontext in der Anfrage AWS KMS übereinstimmt, mit der die Daten verschlüsselt wurden.

Der Geheimtext, den der externe Schlüsselspeicher-Proxy zur Entschlüsselung übermittelt hat, oder der Verschlüsselungskontext ist beschädigt, fehlt oder ist anderweitig ungültig.

- Wenn der vom Proxy empfangene AWS KMS Chiffretext zurückgewiesen wird, bedeutet dies, dass der externe Schlüsselmanager oder der Proxy einen ungültigen oder beschädigten Chiffretext zurückgegeben hat. AWS KMS

  Dieser Fehler kann durch ein Netzwerkproblem während der Übertragung verursacht werden, etwa durch ein „umgekipptes“ Bit. Wiederholen Sie die `Decrypt`-Anforderung. Wenn das Problem weiterhin besteht, stellen Sie sicher, dass der externe Schlüsselmanager ordnungsgemäß funktioniert und dass der externe Schlüsselspeicher-Proxy den Chiffretext, den er vom externen Schlüsselmanager empfängt, nicht ändert, bevor er ihn zurückgibt. AWS KMS

Fehler mit externen Schlüsseln

Ein externer Schlüssel ist ein kryptografischer Schlüssel im externen Schlüsselmanager, der als externes Schlüsselmaterial für einen KMS-Schlüssel dient. AWS KMS kann nicht direkt auf den externen Schlüssel zugreifen. Es muss den externen Schlüsselmanager über den externen Schlüsselspeicher-Proxy bitten, den externen Schlüssel zum Verschlüsseln von Daten oder zum Entschlüsseln eines Geheimtextes zu verwenden.

Sie geben die ID des externen Schlüssels in seinem externen Schlüsselmanager an, wenn Sie einen KMS-Schlüssel in Ihrem externen Schlüsselspeicher erstellen. Sie können die ID des externen Schlüssels nicht mehr ändern, nachdem der KMS-Schlüssel erstellt wurde. Um Probleme mit dem KMS-Schlüssel zu vermeiden, fordert die `CreateKey`-Operation den externen Schlüsselspeicher-Proxy auf, die ID und Konfiguration des externen Schlüssels zu verifizieren. Wenn der externe Schlüssel nicht den Anforderungen für die Verwendung mit einem KMS-Schlüssel entspricht, schlägt die `CreateKey`-Operation mit einer Ausnahme und einer Fehlermeldung fehl, die Aufschluss über das Problem gibt.

Aber auch nach der Erstellung des KMS-Schlüssels können Probleme auftreten. Wenn eine kryptografische Operation aufgrund eines Problems mit dem externen Schlüssel fehlschlägt, schlägt die Operation fehl und gibt eine `KMSInvalidStateException` mit einer Fehlermeldung zurück, die das Problem angibt.

CreateKey Fehler für den externen Schlüssel

Ausnahmen: `XksKeyAlreadyInUseException`, `XksKeyNotFoundException`, `XksKeyInvalidConfigurationException`

Der CreateKeyVorgang versucht, die ID und die Eigenschaften des externen Schlüssels zu überprüfen, den Sie im Parameter Externe Schlüssel-ID (Konsole) oder `XksKeyId` (API) angeben.

Dies dient dazu, Fehler frühzeitig zu erkennen, bevor Sie versuchen, den externen Schlüssel zusammen mit dem KMS-Schlüssel zu verwenden.

Externer Schlüssel wird verwendet

Jeder KMS-Schlüssel in einem externen Schlüsselspeicher muss einen anderen externen Schlüssel verwenden. Wenn `CreateKey` erkannt wird, dass die externe Schlüssel-ID (XksKeyId) für einen KMS-Schlüssel im externen Schlüsselspeicher nicht eindeutig ist, schlägt der Vorgang mit einer fehl`XksKeyAlreadyInUseException`.

Wenn Sie mehrere IDs für denselben externen Schlüssel verwenden, erkennt `CreateKey` das Duplikat nicht. KMS-Schlüssel mit demselben externen Schlüssel sind jedoch nicht interoperabel, da sie unterschiedliche AWS KMS Schlüsselmaterialien und Metadaten haben.

Externer Schlüssel nicht gefunden

Wenn der Proxy für den externen Schlüsselspeicher meldet, dass er den externen Schlüssel mithilfe der externen Schlüssel-ID (XksKeyId) für den KMS-Schlüssel nicht finden kann, schlägt der `CreateKey` Vorgang fehl und es wird die folgende Fehlermeldung angezeigt. `XksKeyNotFoundException`

> Der externe Schlüsselspeicher-Proxy hat die Anforderung abgelehnt, da er den externen Schlüssel nicht finden konnte.

Dieser Fehler kann aus folgenden Gründen auftreten:

- Die ID des externen Schlüssels (`XksKeyId`) für den KMS-Schlüssel ist möglicherweise ungültig. Die ID Ihres externen Schlüssels, den der Proxy zur Identifizierung des externen Schlüssels verwendet, finden Sie in der Dokumentation zu Ihrem externen Schlüsselspeicher-Proxy oder zu Ihrem externen Schlüsselmanager.
- Möglicherweise wurde der externe Schlüssel aus Ihrem externen Schlüsselmanager gelöscht. Verwenden Sie zur Untersuchung dieses Problems die Tools Ihres externen Schlüsselmanagers. Wenn der externe Schlüssel dauerhaft gelöscht wird, verwenden Sie einen anderen externen Schlüssel zusammen mit dem KMS-Schlüssel. Eine Liste der Anforderungen für den externen Schlüssel finden Sie unter Anforderungen an einen KMS-Schlüssel in einem externen Schlüsselspeicher.

Anforderungen an externe Schlüssel nicht erfüllt

Wenn der externe Schlüsselspeicher-Proxy meldet, dass der externe Schlüssel die Anforderungen für die Verwendung mit einem KMS-Schlüssel nicht erfüllt, schlägt die `CreateKey`-Operation fehl und gibteine `XksKeyInvalidConfigurationException` mit einer der folgenden Fehlermeldungen zurück.

Die Schlüsselspezifikation des externen Schlüssels muss AES_256 sein. Die Schlüsselspezifikation des angegebenen externen Schlüssels lautet *<key-spec>* .

- Der externe Schlüssel muss ein symmetrischer 256-Bit-Verschlüsselungsschlüssel mit der Schlüsselspezifikation AES_256 sein. Wenn es sich bei dem angegebenen externen Schlüssel um einen anderen Typ handelt, geben Sie die ID eines externen Schlüssels an, der diese Anforderung erfüllt.

Der Status des externen Schlüssels muss ENABLED (AKTIVIERT) sein. Der Status des angegebenen externen Schlüssels ist *<status>*.

- Der externe Schlüssel muss im externen Schlüsselmanager aktiviert sein. Wenn der angegebene externe Schlüssel nicht aktiviert ist, verwenden Sie die Tools Ihres externen Schlüsselmanagers, um ihn zu aktivieren, oder geben Sie einen aktivierten externen Schlüssel an.

Die Schlüsselnutzung des externen Schlüssels muss ENCRYPT (VERSCHLÜSSELN) und DECRYPT (ENTSCHLÜSSELN) beinhalten. Die Schlüsselnutzung des angegebenen externen Schlüssels ist *<key-usage>* .

- Der externe Schlüssel muss für die Verschlüsselung und Entschlüsselung im externen Schlüsselmanager konfiguriert sein. Wenn der angegebene externe Schlüssel diese Operationen nicht beinhaltet, verwenden Sie die Tools Ihres externen Schlüsselmanagers, um die Operationen zu ändern, oder geben Sie einen anderen externen Schlüssel an.

Kryptografische Operationsfehler für den externen Schlüssel

Ausnahmen: `KMSInvalidStateException`

Wenn der externe Schlüsselspeicher-Proxy den mit dem KMS-Schlüssel verknüpften externen Schlüssel nicht finden kann oder der externe Schlüssel die Anforderungen für die Verwendung mit einem KMS-Schlüssel nicht erfüllt, schlägt die kryptografische Operation fehl.

Probleme mit externen Schlüsseln, die bei einer kryptografischen Operation erkannt werden, sind schwieriger zu beheben als Probleme mit externen Schlüsseln, die vor der Erstellung des KMS-Schlüssels erkannt wurden. Sie können die ID des externen Schlüssels nicht mehr ändern, nachdem der KMS-Schlüssel erstellt wurde. Wenn mit dem KMS-Schlüssel noch keine Daten verschlüsselt wurden, können Sie ihn löschen und einen neuen KMS-Schlüssel mit einer anderen ID für den externen Schlüssel erstellen. Der mit dem KMS-Schlüssel generierte Chiffretext kann jedoch nicht mit einem anderen KMS-Schlüssel entschlüsselt werden, auch nicht mit einem mit demselben externen Schlüssel, da Schlüssel unterschiedliche Schlüsselmetadaten und anderes Schlüsselmaterial haben. AWS KMS Verwenden Sie stattdessen nach Möglichkeit die Tools Ihres externen Schlüsselmanagers, um das Problem mit dem externen Schlüssel zu beheben.

Wenn der externe Schlüsselspeicher-Proxy ein Problem mit dem externen Schlüssel meldet, geben kryptografische Operationen eine `KMSInvalidStateException` mit einer Fehlermeldung zurück, die das Problem identifiziert.

Externer Schlüssel nicht gefunden

Wenn der externe Schlüsselspeicher-Proxy meldet, dass er den externen Schlüssel mithilfe der externen Schlüssel-ID (XksKeyId) für den KMS-Schlüssel nicht finden kann, geben kryptografische Operationen a `KMSInvalidStateException` mit der folgenden Fehlermeldung zurück.

> Der externe Schlüsselspeicher-Proxy hat die Anforderung abgelehnt, da er den externen Schlüssel nicht finden konnte.

Dieser Fehler kann aus folgenden Gründen auftreten:

- Die ID des externen Schlüssels (`XksKeyId`) für den KMS-Schlüssel ist nicht mehr gültig.

  Die ID des externen Schlüssels, der Ihrem KMS-Schlüssel zugeordnet ist, finden Sie in den Details des KMS-Schlüssels. Die ID, die Ihr externer Schlüssel-Proxy zur Identifizierung des externen Schlüssels verwendet, finden Sie in der Dokumentation zu Ihrem externen Schlüsselspeicher-Proxy oder zu Ihrem externen Schlüsselmanager.

AWS KMS überprüft die externe Schlüssel-ID, wenn ein KMS-Schlüssel in einem externen Schlüsselspeicher erstellt wird. Die ID kann jedoch ungültig werden, insbesondere wenn der Wert der ID des externen Schlüssels ein Alias oder ein veränderbarer Name ist. Sie können die einem vorhandenen KMS-Schlüssel zugeordnete ID des externen Schlüssels nicht ändern. Um einen Geheimtext zu entschlüsseln, der unter dem KMS-Schlüssel verschlüsselt wurde, müssen Sie den externen Schlüssel erneut der vorhandenen ID des externen Schlüssels zuordnen.

Wenn Sie den KMS-Schlüssel noch nicht zum Verschlüsseln von Daten verwendet haben, können Sie einen neuen KMS-Schlüssel mit einer gültigen ID des externen Schlüssels erstellen. Haben Sie jedoch mit dem KMS-Schlüssel Geheimtext generiert, können Sie diesen Geheimtext mit keinem anderen KMS-Schlüssel entschlüsseln, selbst wenn derselbe externe Schlüssel verwendet wird.

- Möglicherweise wurde der externe Schlüssel aus Ihrem externen Schlüsselmanager gelöscht. Verwenden Sie zur Untersuchung dieses Problems die Tools Ihres externen Schlüsselmanagers. Versuchen Sie nach Möglichkeit, das Schlüsselmaterial aus einer Kopie oder einem Backup Ihres externen Schlüsselmanagers wiederherzustellen. Wenn der externe Schlüssel dauerhaft gelöscht wird, kann der unter dem zugehörigen KMS-Schlüssel verschlüsselte Geheimtext nicht wiederhergestellt werden.

Fehler bei der Konfiguration externer Schlüssel

Wenn der externe Schlüsselspeicher-Proxy meldet, dass der externe Schlüssel die Anforderungen für die Verwendung mit einem KMS-Schlüssel nicht erfüllt, gibt die kryptografische Operation eine `KMSInvalidStateException` mit einer der folgenden Fehlermeldungen zurück.

Der externe Schlüsselspeicher-Proxy lehnte die Anforderung ab, da der externe Schlüssel die angeforderte Operation nicht unterstützt.

- Der externe Schlüssel muss sowohl die Verschlüsselung als auch die Entschlüsselung unterstützen. Wenn die Schlüsselnutzung keine Verschlüsselung und Entschlüsselung beinhaltet, ändern Sie die Schlüsselnutzung mithilfe der Tools Ihres externen Schlüsselmanagers.

Der externe Schlüsselspeicher-Proxy hat die Anforderung abgelehnt, da der externe Schlüssel im externen Schlüsselmanager nicht aktiviert ist.

- Der externe Schlüssel muss im externen Schlüsselmanager aktiviert und für die Verwendung verfügbar sein. Wenn der Status des externen Schlüssels nicht `Enabled` lautet, aktivieren Sie ihn mithilfe der Tools Ihres externen Schlüsselmanagers.

Proxy-Probleme

Ausnahmen:

`CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (kryptografische Operationen), `UnsupportedOperationException`, `XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (Verwaltungsoperationen außer `CreateKey`)

Der externe Schlüsselspeicher-Proxy vermittelt die gesamte Kommunikation zwischen AWS KMS und dem externen Schlüsselmanager. Er übersetzt generische AWS KMS Anfragen in ein Format, das Ihr externer Schlüsselmanager verstehen kann. Wenn der externe Schlüsselspeicher-Proxy nicht der API-Spezifikation für den AWS KMS externen Schlüsselspeicher-Proxy entspricht oder wenn er nicht ordnungsgemäß funktioniert oder nicht mit ihm kommunizieren kann AWS KMS, können Sie in Ihrem externen Schlüsselspeicher keine KMS-Schlüssel erstellen oder verwenden.

In vielen Fehlermeldungen wird der externe Schlüsselspeicher-Proxy erwähnt, weil er eine wichtige Rolle in der Architektur des externen Schlüsselspeichers spielt. Diese Probleme können ihren Ursprung aber im externen Schlüsselmanager oder im externen Schlüssel haben.

Die Probleme in diesem Abschnitt beziehen sich auf Probleme beim Design oder Betrieb des externen Schlüsselspeicher-Proxys. Zum Beheben dieser Probleme ist möglicherweise eine Änderung an der Proxysoftware erforderlich. Wenden Sie sich an den Administrator für Ihren Proxy. Um Sie bei der Diagnose von Proxy-Problemen zu unterstützen, stellt AWS KMS einen Open-Source-Testclient bereit (XKS-Proxy-Textclient), der überprüft, ob Ihr externer Schlüsselspeicher-Proxy der API-Spezifikation von AWS KMS für externe Schlüsselspeicher-Proxys entspricht.

`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException` oder `XksProxyUriUnreachableException`
Der externe Schlüsselspeicher-Proxy hat einen fehlerhaften Status. Wenn Sie diese Meldung wiederholt sehen, benachrichtigen Sie den Administrator für Ihren externen Schlüsselspeicher-Proxy.

- Dieser Fehler kann auf ein Betriebsproblem oder einen Softwarefehler im externen Schlüsselspeicher-Proxy hindeuten. Sie können CloudTrail Protokolleinträge für den AWS KMS API-Vorgang finden, der die einzelnen Fehler generiert hat. Dieser Fehler kann möglicherweise durch das erneute Ausführen der Operation behoben werden. Sollte er jedoch weiterhin bestehen, benachrichtigen Sie den Administrator für Ihren externen Schlüsselspeicher-Proxy.
- Wenn der externe Schlüsselspeicher-Proxy (in einer GetHealthStatusAntwort) meldet, dass alle externen Schlüsselmanager-Instanzen vorhanden sind`UNAVAILABLE`, schlagen Versuche, einen externen Schlüsselspeicher zu erstellen oder zu aktualisieren, mit dieser Ausnahme fehl. Wenn dieser Fehler weiterhin besteht, lesen Sie in der Dokumentation Ihres externen Schlüsselmanagers nach.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` oder `XksProxyInvalidResponseException`
AWS KMS kann die Antwort des externen Schlüsselspeicher-Proxys nicht interpretieren. Wenn Sie diesen Fehler wiederholt sehen, ziehen Sie den Administrator für Ihren externen Schlüssel speicher-Proxy zurate.

- AWS KMS Operationen erzeugen diese Ausnahme, wenn der Proxy eine undefinierte Antwort zurückgibt, die AWS KMS nicht analysiert oder interpretiert werden kann. Dieser Fehler kann gelegentlich aufgrund vorübergehender externer Probleme oder sporadischer Netzwerkfehler auftreten. Wenn das Problem jedoch weiterhin besteht, kann dies darauf hindeuten, dass der externe Schlüsselspeicher-Proxy nicht der API-Spezifikation von AWS KMS für den externen Schlüsselspeicher-Proxy entspricht. Informieren Sie den Administrator oder Anbieter Ihres externen Schlüsselspeichers.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` oder `UnsupportedOperationException`

Der externe Schlüsselspeicher-Proxy lehnte die Anforderung ab, da er die angeforderte kryptogra fische Operation nicht unterstützt.

- Der externe Schlüsselspeicher-Proxy sollte alle Proxy-APIs unterstützen, die in der API-Spezifikation von AWS KMS für den externen Schlüsselspeicher-Proxy definiert sind. Dieser Fehler

deutet darauf hin, dass der Proxy die Operation, die mit der Anforderung zusammenhängt, nicht unterstützt. Informieren Sie den Administrator oder Anbieter Ihres externen Schlüsselspeichers.

Probleme mit der Proxy-Autorisierung

Ausnahmen: `CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Einige externe Schlüsselspeicher-Proxys implementieren Autorisierungsanforderungen für die Verwendung ihrer externen Schlüssel. Ein externer Schlüsselspeicher-Proxy ist berechtigt, aber nicht verpflichtet, ein Autorisierungsschema zu entwerfen und zu implementieren, das es bestimmten Benutzern erlaubt, bestimmte Operationen unter bestimmten Bedingungen anzufordern. Beispielsweise könnte ein Proxy zulassen, dass ein:e Benutzer:in die Verschlüsselung mit einem bestimmten externen Schlüssel durchführt, aber nicht die Entschlüsselung mit diesem Schlüssel. Weitere Informationen finden Sie unter Proxy-Autorisierung für externen Schlüsselspeicher (optional).

Die Proxyautorisierung basiert auf Metadaten, die in den Anfragen an den Proxy AWS KMS enthalten sind. Die Felder `awsSourceVpc` und `awsSourceVpce` sind nur dann in den Metadaten enthalten, wenn die Anforderung von einem VPC-Endpunkt stammt – und auch nur, wenn der Aufrufer im selben Konto wie der KMS-Schlüssel ist.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Wenn der Proxy eine Anforderung aufgrund eines Autorisierungsfehlers ablehnt, schlägt die zugehörige AWS KMS -Operation fehl. `CreateKey` gibt dann eine `CustomKeyStoreInvalidStateException` zurück. Kryptografische AWS KMS -Operationen geben eine `KMSInvalidStateException` zurück. Beide verwenden die folgende Fehlermeldung:

Der externe Schlüsselspeicher-Proxy hat den Zugriff auf die Operation verweigert. Stellen Sie sicher, dass sowohl der:die Benutzer:in als auch der externe Schlüssel für diese Operation autorisiert ist, und führen Sie die Anforderung erneut aus.

- Um den Fehler zu beheben, ermitteln Sie mithilfe Ihres externen Schlüsselmanagers oder mithilfe der Tools für Ihren externen Schlüsselspeicher-Proxy, warum die Autorisierung fehlgeschlagen ist. Aktualisieren Sie anschließend das Verfahren, das zur unbefugten Anforderung geführt hat, oder verwenden Sie die Tools für Ihren externen Schlüsselspeicher-Proxy, um die Autorisierungsrichtlinie zu aktualisieren. Dieser Fehler lässt sich in AWS KMS nicht beheben.

Schlüsseltypreferenz

AWS KMS unterstützt verschiedene Funktionen für verschiedene Arten von KMS-Schlüsseln. Sie können beispielsweise mit KMS-Schlüsseln zur symmetrischen Verschlüsselung nur symmetrische Datenschlüssel und asymmetrische Datenschlüsselpaare generieren. Außerdem werden das Importieren von Schlüsselmaterial und die automatische Schlüsseldrehung nur für KMS-Schlüssel zur symmetrischen Verschlüsselung unterstützt, und Sie können in einem benutzerdefinierten Schlüsselspeicher nur KMS-Schlüssel zur symmetrischen Verschlüsselung erstellen.

Diese Referenz enthält zwei Tabellen.

- In der Tabelle „Schlüsseltyp“ sind dieAWS KMS Operationen aufgeführt, die für KMS-Schlüssel mit symmetrischer Verschlüsselung, asymmetrische KMS-Schlüssel und HMAC-KMS-Schlüssel gelten.
- In der Tabelle „Spezielle Funktionen“ sind dieAWS KMS Operationen aufgeführt, die für mehrere Regionen geltende KMS-Schlüssel, KMS-Schlüssel mit importiertem Schlüsselmaterial und KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern gültig sind.

Schlüsseltyp-Tabelle

Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in dieser Tabelle anzuzeigen.

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
CancelKeyDeletion	✓	✓	✓	✓

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
CreateAlias	✓	✓	✓	✓
CreateGrant	✓	✓	✓	✓
CreateKey	✓	✓	✓	✓
Decrypt	✓	⊗	✓	⊗
DeleteAlias	✓	✓	✓	✓
DeleteImportedKeyMaterial Gilt nur für KMS-Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`).	✓	✓	✓	✓
DescribeKey	✓	✓	✓	✓
DisableKey	✓	✓	✓	✓

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
DisableKeyRotation	✓ Gilt nur für KMS-Schlüssel mit AWS KMS-Schlüsselmaterial (`Origin` ist `AWS_KMS`).	⊗	⊗	⊗
EnableKey	✓	✓	✓	✓
EnableKeyRotation	✓ Gilt nur für KMS-Schlüssel mit AWS KMS-Schlüsselmaterial (`Origin` ist `AWS_KMS`).	⊗	⊗	⊗

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
Encrypt	✓	⊗	✓	⊗
GenerateDataKey	✓	⊗	⊗	⊗
GenerateDataKeyPair Generierung eines asymmetrischen Datenschlüsselpaars, das durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung geschützt ist.	✓ Gilt nicht für KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern.	⊗	⊗	⊗
GenerateDataKeyPairWithoutPlaintext Generierung eines asymmetrischen Datenschlüsselpaars, das durch einen KMS-Schlüssel mit symmetrischer Verschlüsselung geschützt ist.	✓ Gilt nicht für KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern.	⊗	⊗	⊗
GenerateDataKeyWithoutPlaintext	✓	⊗	⊗	⊗

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
GenerateMac	⊗	✓	⊗	⊗
GetKeyPolicy	✓	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` ist immer `false`.)	✓ (`KeyRotationEnabled` ist immer `false`.)	✓ (`KeyRotationEnabled` ist immer `false`.)
GetParametersForImport Gilt nur für KMS-Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`).	✓	✓	✓	✓
GetPublicKey	⊗	⊗	✓	✓
ImportKeyMaterial Gilt nur für KMS-Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`).	✓	✓	✓	✓
ListAliases	✓	✓	✓	✓

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
ListGrants	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓
ListResourceTags	✓	✓	✓	✓
ListRetirableGrants	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓
ReEncrypt	✓	⊗	✓	⊗
ReplicateKey - gilt nur für multiregionale Schlüssel	✓	✓	✓	✓
RetireGrant	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓	✓
Sign	⊗	⊗	⊗	✓
TagResource	✓	✓	✓	✓

AWS KMS API-Operation	KMS-Schlüssel zur symmetrischen Verschlüsselung	HMAC-KMS-Schlüssel	Asymmetrische KMS-Schlüssel (ENCRYPT_DECRYPT)	Asymmetrische KMS-Schlüssel (SIGN_VERIFY)
UntagResource	✓	✓	✓	✓
UpdateAlias Der aktuelle KMS-Schlüssel und der neue KMS-Schlüssel müssen vom selben Typ sein (beide symmetrisch oder beide asymmetrisch oder beide HMAC) und die gleiche Schlüsselnutzung aufweisen.	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✓	✓
UpdateReplicaRegion - gilt nur für multiregionale Schlüssel	✓	✓	✓	✓
Verify	⊗	⊗	⊗	✓
VerifyMac	⊗	✓	⊗	⊗

Tabelle „Spezielle Funktionen“

Diese Tabelle zeigt die AWS KMS API-Operationen, die für alle Schlüsseltypen für Sonderzwecke unterstützt werden.

Achten Sie beim Lesen dieser Tabelle auf die folgenden Interaktionen:

- Multiregionale Schlüssel:
  - Bei multiregionalen Schlüsseln kann es sich um KMS-Schlüssel mit symmetrischer Verschlüsselung, KMS-Schlüssel mit asymmetrischer Verschlüsselung, HMAC-KMS-Schlüssel und KMS-Schlüssel mit importiertem Schlüsselmaterial handeln.
  - Sie können keine multiregionale Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden.
- Importiertes Schlüsselmaterial
  - Sie können Schlüsselmaterial für symmetrisch verschlüsselte KMS-Schlüssel, asymmetrische KMS-Schlüssel und HMAC-KMS-Schlüssel importieren.
  - Sie können multiregionale Schlüssel mit importiertem Schlüsselmaterial erstellen.
  - Sie können keine Schlüssel mit importiertem Schlüsselmaterial oder in einem benutzerdefinierten Schlüsselspeicher erstellen.
  - Automatische Schlüsseldrehung (`EnableKeyRotation`, `DisableKeyRotation`) wird für KMS-Schlüssel oder KMS-Schlüssel mit importiertem Schlüsselmaterial nicht unterstützt.
- Benutzerdefinierte Schlüsselspeicher
  - Benutzerdefinierte Schlüsselspeicher unterstützen nur KMS-Schlüssel mit symmetrischer Verschlüsselung.
  - Symmetrische Operationen mit asymmetrischen Schlüsselpaaren (`GenerateDataKeyPair`, `GenerateDataKeyPairWithoutPlaintext`) werden für KMS-Schlüssel in benutzerdefinierten Schlüsselspeichern nicht unterstützt.
  - Automatische Schlüsselrotation (`EnableKeyRotation`, `DisableKeyRotation`) wird bei KMS-Schlüsseln in benutzerdefinierten Schlüsselspeichern nicht unterstützt.
  - Sie können keine multiregionalen Schlüssel in benutzerdefinierten Schlüsselspeichern erstellen.

Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in dieser Tabelle anzuzeigen.

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Sie können `CreateKey` verwenden, um einen multiregionalen Primärschlüssel, einen KMS-Schlüssel mit importiertem Schlüsselmaterial oder einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher zu erstellen. Verwenden Sie `ReplicateKey` zum Erstellen eines multiregionalen Replikatschlüssels.	✓	✓	✓
Decrypt	✓ Gilt nur, wenn `KeyUsage` `ENCRYPT_DECRYPT` ist.	✓	✓
DeleteAlias	✓	✓	✓

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
DeleteImportedKeyMaterial	✓ Gilt nur für Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`)	✓	⊗
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓
DisableKeyRotation	✓ Gilt nur für Schlüssel mit symmetrischer Verschlüsselung mit AWS KMS-Schlüsselmaterial (`Origin` ist `AWS_KMS`).	⊗	⊗

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
EnableKey	✓ Gilt nur für KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	✓
EnableKeyRotation	✓ Gilt nur für Schlüssel mit symmetrischer Verschlüsselung mit AWS KMS-Schlüsselmaterial (`Origin` ist `AWS_KMS`).	⊗	⊗
Encrypt	✓ Gilt nur, wenn `KeyUsage` `ENCRYPT_DECRYPT` ist.	✓	✓

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
GenerateDataKey	✓ Gilt nur für KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	✓
GenerateDataKeyPair	✓ Gilt nur für KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	⊗
GenerateDataKeyPairWithoutPlaintext	✓ Gilt nur für KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ Gilt nur für KMS-Schlüssel mit symmetrischer Verschlüsselung	✓	✓

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
GenerateMac Gilt nur für HMAC-KMS-Schlüssel	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` ist immer `false`.)	⊗
GetParametersForImport	✓ Gilt nur für Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`).	✓	⊗
GetPublicKey Gilt nur für asymmetrische KMS-Schlüssel.	✓	✓	⊗

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
ImportKeyMaterial	✓ Gilt nur für Schlüssel mit importiertem Schlüsselmaterial (`Origin` ist `EXTERNAL`).	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
ReEncrypt	✓ Gilt nur, wenn `KeyUsage` `ENCRYPT_DECRYPT` ist.	✓	✓
ReplicateKey	✓ Gilt nur für multiregionale Primärschlüssel.	✓ Gilt nur für multiregionale Primärschlüssel.	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓
Sign Gilt nur, wenn `KeyUsage` `SIGN_VERIFY` ist.	✓	✓	⊗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓

AWS KMS API-Operation	Multiregionale Schlüssel	Importiertes Schlüsselmaterial	KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher
UpdateAlias - Der aktuelle KMS-Schlüssel und der neue KMS-Schlüssel müssen vom selben Typ sein (beide symmetrisch oder beide asymmetrisch oder beide HMAC) und die gleiche Schlüsselnutzung aufweisen.	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ Gilt nur für multiregionale Schlüssel.	⊗
Verify Gilt nur, wenn `KeyUsage` `SIGN_VERIFY` ist.	✓	✓	⊗
VerifyMac Gilt nur für HMAC-KMS-Schlüssel	✓	✓	⊗

Sicherheit von AWS Key Management Service

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die eingerichtet wurde, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der übergreifenden Verantwortlichkeit beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für AWS Key Management Service (AWS KMS) gelten, finden Sie unter AWSDienstleistungen im Umfang des Compliance-Programms.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von AWS Key Management Service einsetzen können. Es zeigt Ihnen, wie Sie AWS KMS konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Themen

Datenschutz in AWS Key Management Service

AWS Key Management Service speichert und schützt Ihre Verschlüsselungsschlüssel, um sie hochverfügbar zu machen und gleichzeitig eine starke und flexible Zugriffssteuerung zu bieten.

Themen

Schutz von Schlüsselmaterial

Standardmäßig generiert und schützt AWS KMS das kryptografische Schlüsselmaterial für KMS-Schlüssel. Darüber hinaus bietet AWS KMS Optionen für Schlüsselmaterial, das außerhalb von AWS KMS erstellt und geschützt wird. Weitere technische Informationen zur KMS-Schlüsseln und Schlüsselmaterial finden Sie unter AWS Key Management Service – Kryptografische Details

Schutz von Schlüsselmaterial, das in AWS KMS generiert wurde

Beim Erstellen eines KMS-Schlüssels generiert und schützt AWS KMS standardmäßig das kryptografische Material für den KMS-Schlüssel.

Um das Schlüsselmaterial für KMS-Schlüssel zu schützen, verlässt AWS KMS sich auf eine verteilte Flotte von Level-3-validierten FIPS-140-2-Hardware-Sicherheitsmodulen (HSMs). Jedes AWS KMS-HSM ist ein dediziertes, eigenständiges Hardware-Gerät, das speziell für die Bereitstellung dedizierter kryptografischer Funktionen zur Erfüllung der Sicherheits- und Skalierbarkeits-Anforderungen von AWS KMS entwickelt wurde. (Die HSMs, die AWS KMS in den chinesischen Regionen einsetzt, sind von OSCCA zertifiziert und entsprechen allen einschlägigen chinesischen Vorschriften, sind aber nicht im Rahmen des FIPS-140-2-Validierungsprogramm für kryptografische Module validiert).

Das Schlüsselmaterial für einen KMS-Schlüssel wird standardmäßig verschlüsselt, wenn er im HSM generiert wird. Das Schlüsselmaterial wird nur innerhalb des flüchtigen HSM-Speichers und nur für die wenigen Millisekunden entschlüsselt, die für die Verwendung in einer kryptografischen Operation benötigt werden. Wenn das Schlüsselmaterial nicht aktiv verwendet wird, wird es innerhalb des HSM verschlüsselt und in einen äußerst dauerhaften (99,999999999 %) persistenten Speicher mit niedriger Latenz übertragen, wo es getrennt und von den HSMs isoliert bleibt. Das Klartext-Schlüsselmaterial verlässt dieHSM-Sicherheitsgrenze nie; es wird nie auf die Festplatte geschrieben

oder in einem Speichermedium gespeichert. (Die einzige Ausnahme ist der öffentliche Schlüssel eines asymmetrischen Schlüsselpaares, der nicht geheim ist).

AWS behauptet als grundlegendes Sicherheitsprinzip, dass es keine menschliche Interaktion mit kryptographischem Schlüsselmaterial im Klartext gibt, egal welcher Art in jedem AWS-Service. Es gibt keinen Mechanismus, mit dem irgendjemand, einschließlich AWS-Service-Betreiber, Schlüsselmaterial im Klartext einsehen, darauf zugreifen oder es exportieren könnte. Dieses Prinzip gilt auch bei katastrophalen Ausfällen und Notfallwiederherstellungsereignissen. Klartext-Kundenschlüsselmaterial in AWS KMS wird für kryptografische Operationen innerhalb von AWS KMS FIPS-validierten HSMs nur als Antwort auf autorisierte Anfragen des Kunden oder seines Beauftragten an den Dienst verwendet.

Bei kundenverwalteten Schlüsseln ist der AWS-Konto, der den Schlüssel erstellt, der alleinige und nicht übertragbare Eigentümer des Schlüssels. Das Eigentümerkonto hat die vollständige und ausschließliche Kontrolle über die Autorisierungsrichtlinien, die den Zugriff auf den Schlüssel regeln. Für Von AWS verwaltete Schlüssel hat das AWS-Konto die vollständige Kontrolle über die IAM-Richtlinien, die Anfragen an das AWS-Service genehmigen.

Schutz von außerhalb von AWS KMS generiertem Schlüsselmaterial

AWS KMS bietet Alternativen zu Schlüsselmaterial, das in AWS KMS generiert wurde.

Mit benutzerdefinierten Schlüsselspeichern, einem optionalen AWS KMS-Feature, können Sie KMS-Schlüssel erstellen, die durch Schlüsselmaterial gesichert sind, das außerhalb von AWS KMS generiert und verwendet wird. KMS-Schlüssel in AWS CloudHSM-Schlüsselspeichern werden durch Schlüssel in AWS CloudHSM-Hardware-Sicherheitsmodulen gesichert, die Sie kontrollieren. Diese HSMs sind FIPS 140-2 Security Level 3 zertifiziert. KMS-Schlüssel in externen Schlüsselspeichern werden durch Schlüssel in einem externen Schlüsselmanager gesichert, den Sie außerhalb von AWS kontrollieren und verwalten, z. B. ein physisches HSM in Ihrem privaten Rechenzentrum.

Ein weiteres optionales Feature ermöglicht das Importieren des Schlüsselmaterials für einen KMS-Schlüssel. Um importiertes Schlüsselmaterial auf dem Weg zu AWS KMS zu schützen, verschlüsseln Sie das Schlüsselmaterial mit einem öffentlichen Schlüssel aus einem RSA-Schlüsselpaar, das in einem AWS KMS-HSM erzeugt wurde. Das importierte Schlüsselmaterial wird in einem AWS KMS-HSM entschlüsselt und mit einem symmetrischen Schlüssel im HSM neu verschlüsselt. Wie jedes AWS KMS-Schlüsselmaterial verlässt das Klartext-Schlüsselmaterial die HSMs niemals unverschlüsselt. Der Kunde, der das Schlüsselmaterial zur Verfügung gestellt hat, ist jedoch verantwortlich für die sichere Verwendung, Haltbarkeit und Wartung des Schlüsselmaterials außerhalb von AWS KMS.

Datenverschlüsselung

Die Daten in AWS KMS bestehen aus AWS KMS keys und dem Schlüsselmaterial des Verschlüsselungsschlüssels, das sie darstellen. Dieses Schlüsselmaterial existiert im Klartext nur innerhalb von AWS KMS-Hardware-Sicherheitsmodulen (HSMs) und nur bei Verwendung. Andernfalls wird das Schlüsselmaterial verschlüsselt und in dauerhaftem persistenten Speicher aufbewahrt.

Das Schlüsselmaterial, das AWS KMS für KMS-Schlüssel generiert, verlässt AWS KMS-HSMs nie unverschlüsselt. Es wird nicht exportiert oder in irgendeiner AWS KMS-API-Operation übertragen. Die Ausnahme ist für multiregionale Schlüssel, wobei AWS KMS einen regionenübergreifenden Replikationsmechanismus verwendet, um das Schlüsselmaterial eines multiregionalen Schlüssels von einem HSM in einer AWS-Region zu einem HSM in einer anderen AWS-Region zu kopieren. Details dazu finden Sie unter Replikationsprozess für Schlüssel mit mehreren Regionen in Kryptografische DetailsAWS Key Management Service.

Themen

Verschlüsselung im Ruhezustand

AWS KMS generiert Schlüsselmaterial für AWS KMS keys in FIPS 140-2 Security Level 3 konforme Hardware-Sicherheitsmodulen (HSMs). Die einzige Ausnahme sind die China-Regionen, in denen die HSMs, die AWS KMS zur Generierung von KMS-Schlüsseln verwendet, allen einschlägigen chinesischen Vorschriften entsprechen, jedoch nicht im Rahmen des FIPS-140-2-Validierungsprogramm für kryptografische Module validiert werden. Bei Nichtgebrauch wird Schlüsselmaterial durch einen HSM-Schlüssel verschlüsselt und in dauerhaftem persistenten Speicher geschrieben. Das Schlüsselmaterial für KMS-Schlüssel und die Verschlüsselungsschlüssel, die das Schlüsselmaterial schützen, verlassen die HSMs niemals in Klartext-Form.

Die Verschlüsselung und Verwaltung von Schlüsselmaterial für KMS-Schlüssel erfolgt vollständig durch AWS KMS.

Weitere Informationen finden Sie unter Arbeiten mit AWS KMS keys unter den kryptografischen Details für AWS Key Management Service.

Verschlüsselung während der Übertragung

Das Schlüsselmaterial, das AWS KMS für KMS-Schlüssel generiert, wird nie exportiert oder in AWS KMS-API-Operationen übertragen. AWS KMS nutzt Schlüsselbezeichner, um die KMS-Schlüssel in API-Operationen darzustellen. In ähnlicher Weise ist Schlüsselmaterial für KMS-Schlüssel in benutzerdefinierten AWS KMS-Schlüsselspeichern nicht exportierbar und wird nie in AWS KMS- oder AWS CloudHSM-API-Operationen übertragen.

Allerdings geben einige AWS KMS-API-Operationen Datenschlüssel zurück. Kunden können außerdem API-Operationen zum Importieren von Schlüsselmaterial für ausgewählte KMS-Schlüssel verwenden.

Alle AWS KMS-API-Aufrufe müssen signiert und mit Transport Layer Security (TLS) übertragen werden. AWS KMS erfordert TLS 1.2 und empfiehlt TLS 1.3 in allen Regionen. AWS KMS unterstützt auch hybrides Post-Quantum-TLS für AWS KMS-Service-Endpunkte in allen Regionen außer in China. AWS KMS unterstützt kein hybrides Post-Quantum-TLS für FIPS-Endpunkte in AWS GovCloud (US). Aufrufe an AWS KMS erfordern auch eine moderne Cipher-Suite, die Perfect Forward Secrecy unterstützt, was bedeutet, dass der Kompromiss eines Geheimnisses, wie z. B. eines privaten Schlüssels, nicht auch den Sitzungsschlüssel gefährdet.

Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Um die Standard-AWS KMS-Endpunkte oder AWS KMS-FIPS-Endpunkte zu verwenden, müssen Clients TLS 1.2 oder höher unterstützen. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2. Eine Liste der AWS KMS-FIPS-Endpunkte finden Sie unter AWS Key Management Service-Endpunkte und -Kontingente im Allgemeine AWS-Referenz.

Die Kommunikation zwischen AWS KMS-Servicehosts und HSMs wird mit Elliptic Curve Cryptography (ECC) und Advanced Encryption Standard (AES) in einem authentifizierten Verschlüsselungsverfahren geschützt. Weitere Informationen finden Sie unter Interne Kommunikationssicherheit in den kryptografischen Details für AWS Key Management Service.

Richtlinie für den Datenverkehr zwischen Netzwerken

AWS KMS unterstützt eine AWS Management Console und einen Satz von API-Operationen, mit denen Sie AWS KMS keys erstellen, verwalten und in kryptografischen Operationen verwenden können.

AWS KMS unterstützt zwei Optionen für die Netzwerk-Konnektivität von Ihrem privaten Netzwerk zu AWS.

- Eine IPsec-VPN-Verbindung über das Internet.
- AWS Direct Connect verknüpft Ihr internes Netzwerk über ein standardmäßiges Ethernet-Glasfaserkabel mit einem AWS Direct Connect-Standort.

Alle AWS KMS-API-Aufrufe müssen signiert und mit Transport Layer Security (TLS) übertragen werden. Die Anrufe erfordern auch eine moderne Verschlüsselungssammlung, die Perfect Forward Secrecy unterstützt. Der Datenverkehr zu den Hardware-Sicherheitsmodulen (HSMs), die Schlüsselmaterial für KMS-Schlüssel speichern, ist nur von bekannten AWS KMS-API-Hosts über das AWS-interne Netzwerk erlaubt.

Um eine direkte Verbindung zu AWS KMS von Ihrer Virtual Private Cloud (VPC) aus herzustellen, ohne Datenverkehr über das öffentliche Internet zu senden, verwenden Sie VPC-Endpunkte, die AWS PrivateLink bereitstellt. Weitere Informationen finden Sie unter Verbindung zu AWS KMS über einen VPC-Endpunkt.

AWS KMS unterstützt auch eine hybride post-Quantum Schlüsselaustauschoption für das Transport Layer Security (TLS)-Netzwerk-Verschlüsselungsprotokoll. Sie können diese TLS-Option verwenden, wenn Sie eine Verbindung zu AWS KMS-API-Endpunkten herstellen.

Identity and Access Management für AWS Key Management Service

Mit AWS Identity and Access Management (IAM) können Sie den Zugriff auf Ihre AWS-Ressourcen sicher kontrollieren. Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, um AWS KMS-Ressourcen zu nutzen. Weitere Informationen finden Sie unter Verwenden von IAM-Richtlinien mit AWS KMS.

Schlüsselrichtlinien sind der primäre Mechanismus zur Steuerung des Zugriffs auf KMS-Schlüssel in AWS KMS. Jeder KMS-Schlüssel muss über eine Schlüsselrichtlinie verfügen. Sie können auch IAM-Richtlinien und Erteilungen, zusammen mit Schlüsselrichtlinien, zum Steuern des Zugriffs auf Ihre KMS-Schlüssel verwenden. Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS.

Wenn Sie eine Amazon Virtual Private Cloud (Amazon VPC) verwenden, können Sie einen Schnittstellen-VPC-Endpunkt erstellen für AWS KMS powered by AWS PrivateLink. Sie können

auch VPC-Endpunktrichtlinien verwenden, um zu bestimmen, welche Prinzipale auf Ihren AWS KMS-Endpunkt zugreifen können, welche API-Aufrufe sie ausführen können und auf welchen KMS-Schlüssel sie zugreifen können. Details hierzu finden Sie unter Steuern des Zugriffs auf einen VPC-Endpunkt.

Protokollieren und Überwachen in AWS Key Management Service

Überwachung ist wichtig, um die Verfügbarkeit, den Status und die Nutzung Ihrer AWS KMS keys in AWS KMS zu verstehen. Überwachung hilft, die Sicherheit, Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS-Lösungen aufrecht zu erhalten. AWS stellt verschiedene Tools für die Überwachung Ihrer KMS-Schlüssel bereit.

AWS CloudTrail-Protokolle

Jeder Aufruf an einen AWS KMS-API-Vorgang wird als Ereignis in einem AWS CloudTrail-Protokoll erfasst. Diese Protokolle zeichnen alle API-Aufrufe von der AWS KMS-Konsole und Aufrufe von AWS KMS und anderen AWS-Services auf. Kontoübergreifende API-Aufrufe, z. B. ein Aufruf zur Verwendung eines KMS-Schlüssels in einem anderen AWS-Konto, werden in den CloudTrail Protokollen beider Konten aufgezeichnet.

Bei der Fehlerbehebung oder Überwachung können Sie mithilfe des Protokolls den Lebenszyklus eines KMS-Schlüssels rekonstruieren. Sie können auch die Verwaltung und Verwendung des KMS-Schlüssels in kryptografischen Operationen anzeigen. Weitere Informationen finden Sie unter the section called “Protokollierung mit AWS CloudTrail”.

Amazon CloudWatch -Protokolle

Überwachen und speichern Sie Ihre Protokolldateien von AWS CloudTrail oder anderen Quellen, und greifen Sie darauf zu. Weitere Informationen finden Sie im Amazon- CloudWatch Benutzerhandbuch.

Für CloudWatch speichert nützliche InformationenAWS KMS, die Ihnen helfen, Probleme mit Ihren KMS-Schlüsseln und den Ressourcen zu vermeiden, die sie schützen. Weitere Informationen finden Sie unter the section called “Überwachung mit CloudWatch”.

Amazon EventBridge

AWS KMS generiert EventBridge Ereignisse, wenn Ihr KMS-Schlüssel gedreht oder gelöscht wird oder das importierte Schlüsselmaterial in Ihrem KMS-Schlüssel abläuft. Suchen Sie nach AWS KMS-Ereignissen (API-Operationen) und leiten sie zum Erfassen von Statusinformationen an eine

oder mehrere Zielfunktionen oder Streams um. Weitere Informationen finden Sie unter the section called “Überwachung mit Amazon EventBridge” und im Amazon- EventBridge Benutzerhandbuch.

Amazon- CloudWatch Metriken

Sie können Ihre KMS-Schlüssel mithilfe von - CloudWatch Metriken überwachen, die Rohdaten von sammeln und AWS KMS in Leistungsmetriken verarbeiten. Die Daten werden in zweiwöchigen Intervallen aufgezeichnet, sodass Sie Trends aktueller und historischer Informationen anzeigen können. Auf diese Weise können Sie verstehen, wie Ihre KMS-Schlüssel verwendet werden und wie sich ihre Verwendung im Laufe der Zeit ändert. Informationen zur Verwendung von CloudWatch Metriken zur Überwachung von KMS-Schlüsseln finden Sie unter AWS KMS Metriken und Dimensionen.

Amazon- CloudWatch Alarme

Überwachen Sie eine Metrik über einen bestimmten, von Ihnen definierten Zeitraum. Der Alarm führt eine oder mehrere Aktionen durch, basierend auf dem Wert der Metrik im Vergleich zu einem bestimmten Schwellenwert in einer Reihe von Zeiträumen. Sie können beispielsweise einen CloudWatch Alarm erstellen, der ausgelöst wird, wenn jemand versucht, einen KMS-Schlüssel zu verwenden, dessen Löschung in einer kryptografischen Operation geplant ist. Dies zeigt an, dass der KMS-Schlüssel noch verwendet wird und wahrscheinlich nicht gelöscht werden sollte. Weitere Informationen finden Sie unter the section called “Erstellen eines Alarms”.

AWS Security Hub

Sie können die Verwendung von AWS KMS auf die Einhaltung von Sicherheitsstandards und bewährten Praktiken mit AWS Security Hub. Security Hub verwendet Sicherheitskontrollen für die Bewertung von Ressourcenkonfigurationen und Sicherheitsstandards, um Sie bei der Einhaltung verschiedener Compliance-Frameworks zu unterstützen. Weitere Informationen finden Sie unter AWS Key Management Service-Steuerungen im AWS Security Hub-Benutzerhandbuch.

Compliance-Validierung für AWS Key Management Service

Die Auditoren Dritter bewerten die Sicherheit und die Compliance von AWS Key Management Service im Rahmen mehrerer AWS-Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Themen

- Compliance-und Sicherheitsdokumente

- Weitere Informationen

Compliance-und Sicherheitsdokumente

Die folgenden Compliance- und Sicherheitsdokumente umfassen AWS KMS. Um sie anzuzeigen, verwenden Sie AWS Artifact.

- Cloud Computing Compliance Controls Catalogue (C5)
- ISO 27001:2013 Erklärung zur Anwendbarkeit (SoA)
- ISO 27001:2013 Zertifizierung
- ISO 27017:2015 Erklärung zur Anwendbarkeit (SoA)
- ISO 27017:2015 Zertifizierung
- ISO 27018:2015 Erklärung zur Anwendbarkeit (SoA)
- ISO 27018:2014 Zertifizierung
- ISO 9001:2015 Zertifizierung
- PCI-DSS-Compliance-Nachweis (AOC) und Zusammenfassung der Verantwortlichkeiten
- Service Organization Controls (SOC)-Bericht 1
- Service Organization Controls (SOC)-Bericht 2
- Service Organization Controls (SOC)-Bericht 2 zur Vertraulichkeit
- FedRAMP-hoch

Weitere Informationen über die Nutzung von AWS Artifact finden Sie unter Herunterladen von Berichten in AWS Artifact.

Weitere Informationen

Ihre Compliance-Verantwortung bei Verwendung von AWS KMS hängt von der Vertraulichkeit der Daten, den Compliance-Zielen des Unternehmens und den geltenden Gesetzen und Vorschriften ab. Wenn Ihre Nutzung von AWS KMS der Einhaltung von einem veröffentlichten Standard ist, stellt AWS Ressourcen zur Unterstützung bereit:

- AWS-Services im Umfang des Compliance Programms – diese Seite listet AWS-Services, die im Umfang von bestimmten Compliance-Programmen enthalten sind. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort interessant sein.
- AWS Config – Dieser AWS-Service bewertet, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- AWS Security Hub Dieser AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb von AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS-Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.

Ausfallsicherheit in AWS Key Management Service

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Zusätzlich zur globalen AWS-Infrastruktur stellt AWS KMS verschiedene Funktionen bereit, um Ihren Anforderungen in Bezug auf Ausfallsicherheit und Datensicherung zu erfüllen. Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWSGlobale Infrastruktur.

Regionale Isolierung

AWS Key Management Service (AWS KMS) ist ein selbsttragender regionaler Dienst, der in allen AWS-Regionen verfügbar ist. Das regional isolierte Design von AWS KMS stellt sicher, dass ein Verfügbarkeitsproblem in einer AWS-Region die AWS KMS-Operation in einer anderen Region nicht beeinträchtigen kann. AWS KMS ist so konzipiert, dass es keine geplanten Ausfallzeiten gibt und alle Software-Updates und Skalierungsvorgänge nahtlos und unmerklich durchgeführt werden.

Das AWS KMS-Service Level Agreement (SLA) beinhaltet eine Serviceverpflichtung von 99,999 % für alle KMS-APIs. Um diese Verpflichtung zu erfüllen, stellt AWS KMS sicher, dass alle Daten

und Berechtigungsinformationen, die zur Ausführung einer API-Anfrage erforderlich sind, auf allen regionalen Hosts verfügbar sind, die die Anfrage erhalten.

Die AWS KMS-Infrastruktur wird in mindestens drei Availability Zones (AZs) in jeder Region repliziert. Um sicherzustellen, dass mehrere Hostausfälle die AWS KMS-Leistung nicht beeinträchtigen, dientAWS KMS dem Kundendatenverkehr von einer der AZs in einer Region.

Änderungen, die Sie an den Eigenschaften oder Berechtigungen eines KMS-Schlüssels vornehmen, werden auf alle Hosts in der Region repliziert, um sicherzustellen, dass nachfolgende Anfragen von jedem Host in der Region korrekt verarbeitet werden können. Anfragen für kryptografische Operationen mit Ihrem KMS-Schlüssel werden an eine Flotte von AWS KMS-Hardware-Sicherheitsmodulen (HSMs) weitergeleitet, von denen jedes die Operation mit dem KMS-Schlüssel durchführen kann.

Design mit mehreren Mandanten

Das Design mit mehreren Mandanten von AWS KMS ermöglicht die Erfüllung des 99,999%igen Verfügbarkeits-SLA und die Aufrechterhaltung hoher Anfrageraten bei gleichzeitigem Schutz der Vertraulichkeit Ihrer Schlüssel und Daten.

Es werden mehrere integritätserzwingende Mechanismen eingesetzt, um sicherzustellen, dass der KMS-Schlüssel, den Sie für die kryptografische Operation angegeben haben, immer verwendet wird.

Das Klartext-Schlüsselmaterial für Ihre KMS-Schlüssel ist umfangreich geschützt. Das Schlüsselmaterial wird im HSM verschlüsselt, sobald es erstellt wird, und das verschlüsselte Schlüsselmaterial wird sofort in einen sicheren Speicher mit geringer Latenzzeit übertragen. Der verschlüsselte Schlüssel wird im HSM abgerufen und entschlüsselt, sobald er benutzt wird. Der Klartextschlüssel verbleibt nur so lange im HSM-Speicher, wie er für den Abschluss der kryptografischen Operation benötigt wird. Dann wird er im HSM erneut verschlüsselt und der verschlüsselte Schlüssel wird wieder gespeichert. Das Klartext-Schlüsselmaterial verlässt die HSMs nie; es wird nie in den permanenten Speicher geschrieben.

Weitere Informationen zu den Mechanismen, die AWS KMS zur Sicherung Ihrer Schlüssel verwendet, finden Sie unter AWS Key Management Service – Kryptographische Details.

Bewährte Methoden der Ausfallsicherheit inAWS KMS

Um die Ausfallsicherheit Ihrer AWS KMS-Ressourcen zu optimieren, sollten Sie die folgenden Strategien in Betracht ziehen.

- Um Ihre Sicherungs- und Notfallwiederherstellungs-Strategie zu unterstützen, sollten Sie multiregionale Schlüssel in Betracht ziehen. Dabei handelt es sich um KMS-Schlüssel, die in einer AWS-Region erstellt und nur in von Ihnen angegebene Regionen repliziert werden. Mit multiregionalen Schlüsseln können Sie verschlüsselte Ressourcen zwischen AWS-Regionen (innerhalb derselben Partition) verschieben, ohne den Klartext preiszugeben, und die Ressource bei Bedarf in einer beliebigen Zielregion entschlüsseln. Zusammengehörige multiregionale Schlüssel sind interoperabel, da sie dasselbe Schlüsselmaterial und dieselbe Schlüssel-ID verwenden, aber sie haben unabhängige Schlüsselrichtlinien für eine hochauflösende Zugangskontrolle. Weitere Informationen zu multiregionalen Schlüsseln finden Sie unter AWS KMS.
- Um Ihre Schlüssel in einem Mehrmandanten-Service wie AWS KMS zu schützen, sollten Sie Zugriffskontrollen verwenden, einschließlich Schlüsselrichtlinien und IAM-Richtlinien. Darüber hinaus können Sie Ihre Anfragen zu AWS KMS über einen VPC-Schnittstellenendpunkt von AWS PrivateLink versenden. Wenn Sie dies tun, erfolgt die gesamte Kommunikation zwischen Ihrer Amazon VPC und AWS KMS vollständig innerhalb des AWS-Netzwerks über einen dedizierten AWS KMS-Endpunkt durchgeführt, der auf Ihre VPC beschränkt ist. Sie können diese Anfragen weiter absichern, indem Sie mit VPC-Endpunktrichtlinien eine zusätzliche Autorisierungsebene schaffen. Weitere Informationen finden Sie unter Verbinden mit AWS KMS über einen VPC-Endpunkt.

Sicherheit der Infrastruktur in AWS Key Management Service

Als verwalteter Service ist AWS Key Management Service (AWS KMS) durch die globalen AWS-Verfahren der Netzwerksicherheit geschützt, die im Whitepaper Amazon Web Services: Übersicht über die Sicherheitsprozesse beschrieben werden.

Um über das Netzwerk auf AWS KMS zuzugreifen, können Sie die AWS KMS-API-Operationen aufrufen, die in der AWS Key Management Service-API-Referenz beschrieben sind. AWS KMS erfordert TLS 1.2 und empfiehlt TLS 1.3 in allen Regionen. AWS KMS unterstützt auch hybrides Post-Quantum-TLS für AWS KMS-Service-Endpunkte in allen Regionen außer in China. AWS KMS unterstützt kein hybrides Post-Quantum-TLS für FIPS-Endpunkte in AWS GovCloud (US). Um die Standard-AWS KMS-Endpunkte oder AWS KMS-FIPS-Endpunkte zu verwenden, müssen Clients TLS 1.2 oder höher unterstützen. Clients müssen außerdem Cipher-Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman (DHE) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Die meisten modernen Systeme, z. B. Java 7 und höher, unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, jedoch unterstützt AWS KMS globale Richtlinienbedingungen, mit denen Sie den Zugriff auf einen KMS-Schlüssel anhand der Quell-IP-Adresse, VPC und des VPC-Endpunkts steuern können. Sie können diese Bedingungsschlüssel in Schlüsselrichtlinien und IAM-Richtlinien verwenden. Diese Bedingungen können jedoch AWS daran verhindern, den KMS-Schlüssel in Ihrem Namen zu verwenden. Details hierzu finden Sie unter AWS globale Bedingungsschlüssel.

Die folgende Schlüsselrichtlinienanweisung erlaubt es beispielsweise Benutzer, die die `KMSTestRole`-Rolle annehmen können, diese AWS KMS key für die angegebenen kryptografischen Operationen zu verwenden, es sei denn, die Quell-IP-Adresse ist eine der in der Richtlinie angegebenen IP-Adressen.

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

Isolierung auf physischen Hosts

Die Sicherheit der physischen Infrastruktur, die AWS KMS verwendet, unterliegt den Kontrollen, die im Abschnitt Physische und ökologische Sicherheit in Amazon Web Services: Übersicht der Sicherheitsverfahren beschrieben sind. Weitere Details finden Sie in Compliance-Berichten und Prüfungsergebnissen von Drittanbietern, die im vorherigen Abschnitt aufgeführt sind.

AWS KMS wird von dedizierten Hardware-Sicherheitsmodulen (HSMs) unterstützt, die mit speziellen Steuerelementen gegen physische Angriffe ausgelegt sind. Die HSMs sind physische Geräte, die keine Virtualisierungsebene, z. B. einen Hypervisor, haben, die das physische Gerät an mehrere logische Mandanten freigibt. Das Schlüsselmaterial für AWS KMS keys wird nur im flüchtigen Speicher auf den HSMs gespeichert und nur während der KMS-Schlüssel verwendet wird. Dieser Speicher wird gelöscht, wenn das HSM den Betriebszustand verlässt, einschließlich beabsichtigtem und unbeabsichtigtem Herunterfahren und Zurücksetzen. Ausführliche Informationen zum Betrieb von AWS KMS-HSMs finden Sie unter Kryptografische Details für AWS Key Management Service.

Bewährte Methoden für die Sicherheit für AWS Key Management Service

AWS Key Management Service(AWS KMS) unterstützt viele Sicherheitsfunktionen, die Sie implementieren können, um den Schutz Ihrer Verschlüsselungsschlüssel zu verbessern, einschließlichSchlüsselrichtlinienundIAM-Richtlinien, einVerschlüsselungskontextOption für kryptografische Operationen an symmetrischen Verschlüsselungsschlüsseln, ein umfangreicher Satz von-Bedingungsschlüsselum Ihre Schlüsselrichtlinien und IAM-Richtlinien zu verfeinern,Einschränkungen für Erteilungenum Zuschüsse zu begrenzen.

Diese Funktionen werden unter AWS Key Management Service – Bewährte Methoden ausführlich beschrieben. Diese bewährten Methoden stellen allgemeine Leitlinien dar und bilden keine vollständige Sicherheitslösung. Da nicht alle bewährten Methoden für alle Situationen geeignet sind, sollten diese nicht vorschriftig sein.

Informationen finden Sie auch unter:

- Bewährte Methoden für IAM-Richtlinien
- Bewährte Methoden für AWS KMS-Erteilungen
- Bewährte Methoden für die Sicherheit im IAM-Benutzerhandbuch.

Kontingente

AWS KMS Wendet zwei Arten von Kontingenten an: Ressourcenkontingente und Anforderungskontingente, um für alle Benutzer AWS KMS reaktionsschnell und performant zu sein. Jedes Kontingent wird unabhängig für jede Region jedes AWS-Konto berechnet.

Alle AWS KMS Kontingente sind anpassbar, mit Ausnahme des Ressourcenkontingents für wichtige Richtliniendokumente, des Ressourcenkontingents für die On-Demand-Rotation und des Kontingents für AWS CloudHSM wichtige Speicheranforderungen. Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch. Um eine Senkung des Kontingents zu beantragen, ein Kontingent zu ändern, das nicht in Service Quotas aufgeführt ist, oder um ein Kontingent AWS-Region in einem Bereich zu ändern, für AWS KMS den Servicekontingente nicht verfügbar sind, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

Themen

Ressourcenkontingente

AWS KMS legt Ressourcenkontingente fest, um sicherzustellen, dass es allen unseren Kunden einen schnellen und zuverlässigen Service bieten kann. Einige Ressourcenkontingente gelten nur für Ressourcen, die Sie erstellen, nicht jedoch für Ressourcen, die AWS Dienste für Sie erstellen. Ressourcen, die Sie verwenden, die sich jedoch nicht in ihrem AWS-Konto befinden, z. B. AWS-eigene Schlüssel, werden auf diese Kontingente nicht angerechnet.

Wenn Sie ein Ressourcenlimit erreicht haben, generieren Anforderungen zum Erstellen einer zusätzlichen Ressource dieses Typs die Fehlermeldung `LimitExceededException`.

Alle AWS KMS Ressourcenkontingente sind anpassbar, mit Ausnahme der Größenkontingente für wichtige Richtliniendokumente und der Ressourcenkontingente für die On-Demand-Rotation. Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch. Um eine Senkung des Kontingents zu beantragen, ein Kontingent zu ändern, das nicht in Service Quotas aufgeführt ist, oder um ein Kontingent AWS-

Region in einem Bereich zu ändern, für AWS KMS den Servicekontingente nicht verfügbar sind, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

In der folgenden Tabelle sind die AWS KMS Ressourcenkontingente in den einzelnen Regionen aufgeführt AWS-Konto und beschrieben.

Kontingentname	Standardwert	Gilt für	Einstellbar
AWS KMS keys	100 000	Kundenverwaltete Schlüssel	Ja
Aliasse pro KMS-Schlüssel	50	Kundenseitig erstellte Aliasse	Ja
Erteilungen pro KMS-Schlüssel	50 000	Kundenverwaltete Schlüssel	Ja
Größe des Schlüssel richtliniendokuments	32 KB (32.768 Bytes)	Kundenverwaltete Schlüssel Von AWS verwaltete Schlüssel	Nein
Ressourcenkontingent für benutzerdefinierte Schlüsselspeicher	10	AWS-Konto und Region	Ja

AWS KMS Verwendet zusätzlich zu den Ressourcenkontingenten Anforderungskontingente, um die Reaktionsfähigkeit des Dienstes sicherzustellen. Details hierzu finden Sie unter the section called “Anforderungskontingente”.

AWS KMS keys: 100.000

In jeder Region Ihres AWS-Konto sind bis zu 100.000 kundenverwaltete Schlüssel zulässig. Dieses Kontingent gilt für alle vom Kunden verwaltete Schlüssel in allen AWS-Regionen , unabhängig von ihrer Schlüsselspezifikation oder ihrem Schlüsselstatus. Jeder KMS-Schlüssel wird als eine Ressource betrachtet. Von AWS verwaltete Schlüssel und AWS-eigene Schlüssel werden nicht auf dieses Kontingent angerechnet.

Aliasse pro KMS-Schlüssel: 50

Sie können jedem kundenverwalteten Schlüssel bis zu 50 Aliasse zuordnen. Aliase, die mit AWS verknüpft sind, werden Von AWS verwaltete Schlüsselnicht auf dieses Kontingent angerechnet. Dieses Kontingent kann auftreten, wenn Sie ein Alias erstellen oder aktualisieren.

> **Note**
>
> Die ResourceAliases Bedingung kms: ist nur wirksam, wenn der KMS-Schlüssel diesem Kontingent entspricht. Wenn ein KMS-Schlüssel dieses Kontingent überschreitet, wird auch Prinzipalen, die berechtigt sind, den KMS-Schlüssel zu nutzen, durch die Bedingung `kms:ResourceAliases` der Zugriff auf den KMS-Schlüssel verweigert. Details hierzu finden Sie unter Zugriff aufgrund eines Aliaskontingents verweigert.

Das Schlüsselkontingent Aliase pro KMS ersetzt das Kontingent Aliase pro Region, das die Gesamtzahl der Aliase in jeder Region eines beschränkte. AWS-Konto AWS KMS hat das Kontingent „Aliase pro Region“ gestrichen.

Erteilungen pro KMS-Schlüssel: 50 000

Jedem kundenverwalteten Schlüssel können bis zu 50 000 Erteilungen zugewiesen werden, einschließlich der Erteilungen, die von in AWS KMS integrierten AWS -Services erstellt wurden. Dieses Kontingent gilt nicht für Von AWS verwaltete Schlüssel oder AWS-eigene Schlüssel.

Eine Auswirkung dieses Kontingents besteht darin, dass Sie gleichzeitig nicht mehr als 50 000 per Erteilung autorisierte Operationen durchführen können, die denselben KMS-Schlüssel verwenden. Nachdem Sie das Kontingent erschöpft haben, können Sie neue Erteilungen für den KMS-Schlüssel erst erstellen, nachdem eine aktive Erteilung aufgehoben oder widerrufen wird.

Wenn Sie beispielsweise ein Amazon-Elastic-Block-Store-Volume (Amazon EBS) an eine Amazon-Elastic-Compute-Cloud-Instance (Amazon EC2) anfügen, wird das Volume entschlüsselt, sodass Sie es lesen können. Um die Berechtigung zum Entschlüsseln der Daten zu erhalten, erstellt Amazon EBS für jedes Volume eine Erteilung. Wenn also alle Ihre Amazon-EBS-Volumes denselben KMS-Schlüssel verwenden, können Sie nicht mehr als 50 000 Volumes gleichzeitig anhängen.

Größe des Schlüsselrichtliniendokuments: 32 KB

Die maximale Länge der einzelnen Schlüsselrichtliniendokumente ist 32 KB (32,768 Bytes). Wenn Sie ein größeres Richtliniendokument zum Erstellen oder Aktualisieren der Schlüsselrichtlinie für einen KMS-Schlüssel verwenden, schlägt die Operation fehl.

Dieses Kontingent ist nicht anpassbar. Sie können es nicht erhöhen, indem Sie Servicekontingente verwenden oder einen Fall in erstellen AWS Support. Wenn sich Ihre Schlüsselrichtlinie dem Limit nähert, sollten Sie sich überlegen, Erteilungen anstelle von Richtlinienanweisungen zu nutzen. Erteilungen eignen sich besonders gut für temporäre oder sehr spezifische Berechtigungen.

Sie verwenden ein Dokument mit wichtigen Richtlinien immer dann, wenn Sie eine wichtige Richtlinie erstellen oder ändern, indem Sie die Standardansicht oder Richtlinienansicht in der AWS Management Console oder der PutKeyPolicyOperation verwenden. Dieses Kontingent gilt für das Schlüsselrichtliniendokument, auch wenn Sie die Standardansicht in der AWS KMS -Konsole verwenden, in der Sie die JSON-Anweisungen nicht direkt bearbeiten.

Ressourcenkontingent für benutzerdefinierte Schlüsselspeicher: 10

Sie können in jeder AWS-Konto Region bis zu 10 benutzerdefinierte Schlüsselspeicher erstellen. Wenn Sie versuchen, weitere zu erstellen, schlägt der CreateCustomKeyStoreVorgang fehl.

Dieses Kontingent gilt für die Gesamtzahl der benutzerdefinierten Schlüsselspeicher in jedem Konto und jeder Region, einschließlich aller AWS CloudHSM -Schlüsselspeicher und externer Schlüsselspeicher, unabhängig von ihrem Verbindungsstatus.

Rotation auf Anfrage: 10

Sie können die Schlüsselrotation bei Bedarf maximal 10 Mal pro KMS-Schlüssel durchführen. Wenn Sie versuchen, mehr Rotationen auf Anforderung durchzuführen, schlägt der RotateKeyOnDemandVorgang fehl.

Dieses Kontingent ist nicht anpassbar. Sie können es nicht erhöhen, indem Sie Servicekontingente verwenden oder einen Fall in erstellen AWS Support. Um zu verhindern, dass das Kontingent für die On-Demand-Rotation erreicht wird, empfehlen wir, wann immer möglich die automatische Schlüsselrotation zu verwenden.

Anforderungskontingente

AWS KMS legt Kontingente für die Anzahl der in jeder Sekunde angeforderten API-Operationen fest. Die Anforderungskontingente unterscheiden sich je nach API-Vorgang AWS-Region, dem und anderen Faktoren, wie dem KMS-Schlüsseltyp. Wenn Sie ein API-Anforderungskontingent überschreiten, wird die Anfrage AWS KMS gedrosselt.

Alle AWS KMS Anforderungskontingente sind einstellbar, mit Ausnahme des Anforderungskontingents für AWS CloudHSM Schlüsselspeicher. Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch. Um eine Senkung des Kontingents zu beantragen, ein Kontingent zu ändern, das nicht in Service Quotas aufgeführt ist, oder um ein Kontingent AWS-Region in einem Bereich zu ändern, für AWS KMS den Servicekontingente nicht verfügbar sind, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

Wenn Sie das Anforderungskontingent für den GenerateDataKeyVorgang überschreiten, sollten Sie in Erwägung ziehen, die Funktion zum Zwischenspeichern von Datenschlüsseln zu verwenden. AWS Encryption SDK Durch die Wiederverwendung von Datenschlüsseln kann sich die Häufigkeit Ihrer Anfragen auf verringern. AWS KMS

Zusätzlich zu den Anforderungskontingenten werden Ressourcenkontingente AWS KMS verwendet, um die Kapazität für alle Benutzer sicherzustellen. Details hierzu finden Sie unter Ressourcenkontingente.

Um Trends in Ihren Anforderungsraten anzuzeigen, verwenden Sie die Service-Quotas-Konsole. Sie können auch einen CloudWatchAmazon-Alarm einrichten, der Sie benachrichtigt, wenn Ihre Anforderungsrate einen bestimmten Prozentsatz eines Kontingents erreicht. Einzelheiten finden Sie im AWS Sicherheitsblog unter Verwalten Sie Ihre AWS KMS API-Anforderungsraten mithilfe von Service Quotas und Amazon CloudWatch.

Themen

Fordern Sie Kontingente für jeden AWS KMS API-Vorgang an

In dieser Tabelle sind der Quota-Code für Service-Kontingente und der Standardwert für jedes AWS KMS Anforderungskontingent aufgeführt. Alle AWS KMS Anforderungskontingente sind einstellbar, mit Ausnahme des Anforderungskontingents für AWS CloudHSM Schlüsselspeicher.

> **Note**
>
> Möglicherweise müssen Sie horizontal oder vertikal Scrollen, um alle Daten in dieser Tabelle anzuzeigen.

Kontingentname	Standardwert (Anforderungen pro Sekunde)

`Cryptographic operations (symmetric) request rate`
 	
Gilt für:
 	
• `Decrypt`
• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
• `GenerateMac`
• `GenerateRandom`
• `ReEncrypt`
• `VerifyMac`
	Diese gemeinsamen Kontingente variieren je nach dem AWS-Region und dem Typ des in der Anfrage verwendeten KMS-Schlüssels. Jedes Kontingent wird separat berechnet.
	
	• 5 500 (freigegeben)
	• 10 000 (gemeinsam) in den folgenden Regionen:
	  • USA Ost (Ohio), us-east-2
	  • Asien-Pazifik (Singapur), ap-southeast-1
	  • Asien-Pazifik (Sydney), ap-southeast-2
	  • Asien-Pazifik (Tokio), ap-northeast-1
	  • Europa (Frankfurt) eu-central-1
	  • Europa (London) eu-west-2
	• 50 000 (gemeinsam) in den folgenden Regionen:
	  • USA Ost (Nord-Virginia), us-east-1
	  • USA West (Oregon), us-west-2
	  • Europa (Irland), eu-west-1

Kontingentname	Standardwert (Anforderungen pro Sekunde)
`Cryptographic operations (RSA) request rate` Gilt für: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	500 (gemeinsam) für RSA-KMS-Schlüssel
`Cryptographic operations (ECC and SM2) request rate` Gilt für: • `Decrypt`— wird nur für SM2-KMS-Schlüssel (nur Regionen Chinas) unterstützt • `Encrypt`— wird nur für SM2-KMS-Schlüssel (nur Regionen Chinas) unterstützt • `ReEncrypt` — wird nur für SM2-KMS-S chlüssel (nur Regionen Chinas) unterstützt • `Sign` • `Verify`	300 (gemeinsam genutzt) für KMS-Schlüssel mit elliptischen Kurven (ECC) und SM2 (nur Regionen Chinas)

Kontingentname	Standardwert (Anforderungen pro Sekunde)
`Custom key store request quotas` Gilt für: • `Decrypt` • `Encrypt` • `GenerateDataKey` • `GenerateDataKeyWithoutPlaintext` • `GenerateRandom` • `ReEncrypt`	Anforderungskontingente für benutzerdefinierte Schlüsselspeicher werden für jeden benutzerdefinierten Schlüsselspeicher separat berechnet. • 1.800 (gemeinsam genutzt) für jeden Schlüsselspeicher AWS CloudHSM • 1 800 (freigegeben) für jeden externen Schlüsselspeicher.
`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`	5
`DescribeCustomKeyStores request rate`	5
`DescribeKey request rate`	2000

Kontingentname	Standardwert (Anforderungen pro Sekunde)
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 Gilt für: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P384) request rate Gilt für: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P521) request rate Gilt für: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

Kontingentname	Standardwert (Anforderungen pro Sekunde)
`GenerateDataKeyPair (ECC_SECG_P256K1) request rate` Gilt für: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (RSA_2048) request rate` Gilt für: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	1
`GenerateDataKeyPair (RSA_3072) request rate` Gilt für: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,5 (1 in jedem 2-Sekunden-Intervall)
`GenerateDataKeyPair (RSA_4096) request rate` Gilt für: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,1 (1 in jedem 10-Sekunden-Intervall)

Kontingentname	Standardwert (Anforderungen pro Sekunde)
`GenerateDataKeyPair (SM2 – China Regions only) request rate` Gilt für: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	25
`GetKeyPolicy request rate`	1000
`GetKeyRotationStatus request rate`	1000
`GetParametersForImport request rate`	0,25 (1 in jedem 4-Sekunden-Intervall)
`GetPublicKey request rate`	2000
`ImportKeyMaterial request rate`	5
`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

Kontingentname	Standardwert (Anforderungen pro Sekunde)
`ReplicateKey request rate` Eine `ReplicateKey` -Produktion zählt als eine `ReplicateKey` -Anforderung in der Region des Primärschlüssels und zwei `CreateKey` -Anforderungen in der Region des Replikats. Eine der `CreateKey` -Anforder ungen ist ein Trockenlauf, um potenzielle Probleme zu erkennen, bevor der Schlüssel erstellt wird.	5
`RetireGrant request rate`	30
`RevokeGrant request rate`	30
`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` Eine `UpdatePrimaryRegion` -Produktion zählt als zwei `UpdatePrimaryRegion` - Anforderungen; eine Anforderung in jeder der beiden betroffenen Regionen.	5

Anwenden von Anforderungskontingenten

Beachten Sie beim Anzeigen der Anforderungskontingente die folgenden Informationen.

- Anforderungskontingente gelten sowohl für kundenverwaltete Schlüssel als auch für Von AWS verwaltete Schlüssel. Die Verwendung von AWS-eigene Schlüsselwird nicht auf Ihre Anforderungskontingente angerechnet AWS-Konto, auch wenn sie zum Schutz der Ressourcen in Ihrem Konto verwendet werden.
- Anforderungskontingente gelten für Anforderungen, die an FIPS-Endpunkte und Nicht-FIPS-Endpunkte gesendet werden. Eine Liste der AWS KMS Dienstendpunkte finden Sie unter AWS Key Management Service Endpunkte und Kontingente in der. Allgemeine AWS-Referenz
- Die Drosselung basiert auf allen Anforderungen an KMS-Schlüssel aller Typen in der Region. Diese Summe beinhaltet Anfragen von allen Principals in der AWS-Konto, einschließlich Anfragen von AWS Diensten in Ihrem Namen.
- Jedes Anforderungskontingent wird separat berechnet. Anfragen für den CreateKeyVorgang haben beispielsweise keine Auswirkung auf das Anforderungskontingent für den CreateAliasVorgang. Wenn Ihre `CreateAlias`-Anforderungen gedrosselt werden, können Ihre `CreateKey`-Anforderungen weiterhin erfolgreich abgeschlossen werden.
- Obwohl für kryptografische Operationen ein gemeinsames Kontingent gilt, wird das gemeinsame Kontingent unabhängig von den Kontingenten für andere Operationen berechnet. Beispielsweise teilen sich Aufrufe der Vorgänge Verschlüsseln und Entschlüsseln ein Anforderungskontingent, aber dieses Kontingent ist unabhängig von dem Kontingent für Verwaltungsvorgänge, wie z. EnableKey Beispielsweise können Sie in der Region Europa (London) 10 000 kryptografische Operationen für symmetrische KMS-Schlüssel plus 5 `EnableKey`-Operationen pro Sekunde ausführen, ohne gedrosselt zu werden.

Gemeinsame Kontingente für kryptografische Operationen

AWS KMS Bei kryptografischen Vorgängen werden Anforderungsquoten gemeinsam genutzt. Sie können eine beliebige Kombination der vom KMS-Schlüssel unterstützten kryptografischen Operationen anfordern. Dabei darf nur die Gesamtzahl der kryptografischen Operationen das Anforderungskontingent für diesen KMS-Schlüsseltyp nicht überschreiten. Die Ausnahmen sind GenerateDataKeyPairund GenerateDataKeyPairWithoutPlaintext, die sich ein separates Kontingent teilen.

Die Kontingente für unterschiedliche Schlüsseltypen werden ebenfalls unabhängig berechnet. Jedes Kontingent gilt für alle Anfragen für diese Operationen in der Region AWS-Konto und mit dem angegebenen Schlüsseltyp in jedem Intervall von einer Sekunde.

- Die Anforderungsrate für kryptographische Operationen (symmetrisch) ist das für kryptographische Operationen freigegebene Anforderungskontingent unter Verwendung symmetrischer KMS-Schlüssel in einem Konto und in einer Region. Dieses Kontingent gilt für kryptografische Operationen mit symmetrischen Verschlüsselungsschlüsseln und HMAC-Schlüsseln, die ebenfalls symmetrisch sind.

  Beispielsweise könnten Sie symmetrische KMS-Schlüssel in einem System AWS-Region mit einem gemeinsamen Kontingent von 10.000 Anfragen pro Sekunde verwenden. Wenn Sie 7.000 GenerateDataKeyAnfragen pro Sekunde und 2.000 Decrypt-Anfragen pro Sekunde stellen, werden Ihre Anfragen AWS KMS nicht gedrosselt. Wenn jedoch 9 500 `GenerateDataKey`-Anforderungen und 1 000 Encrypt-Anforderungen pro Sekunde anfallen, drosselt AWS KMS die Anforderungen, da sie das gemeinsame Kontingent überschreiten.

  Kryptografische Operationen mit den KMS-Schlüsseln für symmetrische Verschlüsselung in einem benutzerdefinierten Schlüsselspeicher werden sowohl auf die Anforderungsrate kryptografischer Operationen (symmetrisch) für das Konto als auch auf das Anforderungskontingent für benutzerdefinierte Schlüsselspeicher für den benutzerdefinierten Schlüsselspeicher angerechnet.
- Die Anforderungsrate für kryptografische Operationen (RSA) ist das für kryptografische Operationen freigegebene Anforderungskontingent unter Verwendung von asymmetrischen RSA-KMS-Schlüssel.

  Bei einem Anforderungskontingent von 500 Operationen pro Sekunde können Sie beispielsweise 200 Encrypt-Anforderungen und 100 Decrypt-Anforderungen mit RSA-KMS-Schlüsseln senden, die verschlüsseln und entschlüsseln können, sowie 50 Sign-Anforderungen und 150 Verify-Anforderungen mit RSA-KMS-Schlüsseln, die signieren und verifizieren können.
- Die Anforderungsrate für kryptografische Operationen (ECC) ist das freigegebene Anforderungskontingent für kryptografische Operationen unter Verwendung von asymmetrischen Elliptic-Curve-(ECC)-KMS-Schlüsseln.

  Bei einem Anforderungskontingent von 300 Operationen pro Sekunde können Sie beispielsweise 100 Signierungsanforderungen und 200 Verifizierungsanforderungen mit RSA-KMS-Schlüsseln senden, die signieren und verifizieren können.

- Die Anforderungsrate für kryptografische Operationen (SM – nur China-Regionen) ist das für kryptografische Operationen freigegebene Anforderungskontingent unter Verwendung von asymmetrischen SM-KMS-Schlüssel.

  Bei einem Anforderungskontingent von 300 Operationen pro Sekunde können Sie beispielsweise 100 Encrypt-Anforderungen und 100 Decrypt-Anforderungen mit SM2-KMS-Schlüsseln senden, die verschlüsseln und entschlüsseln können, sowie 50 Sign-Anforderungen und 50 Verify-Anforderungen mit SM2-KMS-Schlüsseln, die signieren und überprüfen können.
- Das Anforderungskontingent für KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher ist das gemeinsame Anforderungskontingent für kryptografische Operationen mit KMS-Schlüsseln in einem benutzerdefinierten Schlüsselspeicher. Dieses Kontingent wird für jeden benutzerdefinierten Schlüsselspeicher separat berechnet.

  Kryptografische Operationen mit den KMS-Schlüsseln für symmetrische Verschlüsselung in einem benutzerdefinierten Schlüsselspeicher werden sowohl auf die Anforderungsrate kryptografischer Operationen (symmetrisch) für das Konto als auch auf das Anforderungskontingent für benutzerdefinierte Schlüsselspeicher für den benutzerdefinierten Schlüsselspeicher angerechnet.

Die Kontingente für unterschiedliche Schlüsseltypen werden ebenfalls unabhängig berechnet. Wenn Sie in der Region Asien-Pazifik (Singapur) beispielsweise symmetrische und asymmetrische KMS-Schlüssel verwenden, können Sie bis zu 10 000 Aufrufe pro Sekunde mit symmetrischen KMS-Schlüsseln (einschließlich HMAC-Schlüsseln) plus bis zu 500 zusätzliche Aufrufe pro Sekunde mit Ihren asymmetrischen RSA-KMS-Schlüsseln plus bis zu 300 zusätzliche Anforderungen pro Sekunde mit Ihren ECC-basierten KMS-Schlüsseln ausgeben.

API-Anforderungen in Ihrem Namen

Sie können API-Anfragen direkt oder mithilfe eines integrierten AWS Dienstes stellen, der API-Anfragen in AWS KMS Ihrem Namen stellt. Das Kontingent gilt für beide Arten von Anforderungen.

Sie speichern Daten beispielsweise in Simple Storage Service (Amazon S3) und verwenden serverseitige Verschlüsselung mit einem KMS-Schlüssel (SSE-KMS). Jedes Mal, wenn Sie ein mit SSE-KMS verschlüsseltes S3-Objekt hoch- oder herunterladen, stellt Amazon S3 in Ihrem `GenerateDataKey` Namen eine Anfrage (für Uploads) oder `Decrypt` (für Downloads) AWS KMS an. Diese Anfragen werden auf Ihr Kontingent angerechnet. Das bedeutet, dass die Anfragen AWS KMS gedrosselt werden, wenn Sie insgesamt 5.500 (oder 10.000 oder 50.000, je nach Ihren Anforderungen AWS-Region) Uploads oder Downloads von mit SSE-KMS verschlüsselten S3-Objekten pro Sekunde überschreiten.

Kontoübergreifende Anforderungen

Wenn eine Anwendung in einer Anwendung einen KMS-Schlüssel AWS-Konto verwendet, der einem anderen Konto gehört, spricht man von einer kontoübergreifenden Anfrage. Bei kontoübergreifenden Anforderungen drosselt AWS KMS das Konto, das die Anforderungen sendet, nicht das Konto, das den KMS-Schlüssel besitzt. Wenn beispielsweise eine Anwendung in Konto A einen KMS-Schlüssel in Konto B verwendet, wird die KMS-Schlüsselnutzung nur auf die Kontingente in Konto A angerechnet.

Anforderungskontingente für benutzerdefinierte Schlüsselspeicher

AWS KMS verwaltet Anforderungskontingente für kryptografische Operationen mit den KMS-Schlüsseln in einem benutzerdefinierten Schlüsselspeicher. Diese Anforderungskontingente werden für jeden benutzerdefinierten Schlüsselspeicher separat berechnet.

Anforderungskontingent für benutzerdefinierte Schlüsselspeicher	Standardwert (Anforderungen pro Sekunde) für jeden benutzerdefinierten Schlüsselspeicher	Einstellbar
AWS CloudHSM Kontingent für Schlüsselspeicher-Anfragen	1800	Nein
Anforderungskontingent für externen Schlüsselspeicher	1800	Ja

> **Note**
>
> AWS KMS benutzerdefinierte Schlüsselspeicher-Anforderungskontingente werden nicht in der Service Quotas Quotas-Konsole angezeigt. Sie können diese Kontingente nicht mithilfe von Service-Quotas-API-Vorgängen anzeigen oder verwalten. Um eine Änderung Ihres Anforderungskontingents für externe Schlüsselspeicher zu beantragen, erstellen Sie im AWS Support Center einen Fall.
>
> Wenn der einem AWS CloudHSM Schlüsselspeicher zugeordnete AWS CloudHSM Cluster zahlreiche Befehle verarbeitet, auch solche, die nichts mit dem benutzerdefinierten Schlüsselspeicher zu tun haben, erhalten Sie möglicherweise einen AWS KMS

`ThrottlingException` lower-than-expected AT-Schlüssel. Wenn dies der Fall ist, reduzieren Sie Ihre Anforderungsrate auf AWS KMS, reduzieren Sie die damit verbundene Last oder verwenden Sie einen dedizierten AWS CloudHSM Cluster für Ihren AWS CloudHSM Schlüsselspeicher.
AWS KMS meldet die Drosselung von externen Schlüsselspeicher-Anfragen in der Metrik. `ExternalKeyStoreThrottle` CloudWatch Sie können diese Metrik verwenden, um Drosselungsmuster anzuzeigen, Alarme zu erstellen und Ihr Kontingent für externe Schlüsselspeicheranforderungen anzupassen.

Eine Anforderung einer kryptografischen Operation zu einem KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher wird auf zwei Kontingente angerechnet:

- Anforderungsrate für kryptografische Operationen (symmetrisch. pro Konto)

  Anfragen für kryptografische Operationen mit KMS-Schlüsseln in einem benutzerdefinierten Schlüsselspeicher werden auf das `Cryptographic operations (symmetric) request rate`-Kontingent für jedes AWS-Konto und jede Region angerechnet. In der Region USA Ost (Nord-Virginia) (us-east-1) zum Beispiel kann jedes AWS-Konto bis zu 50 000 Anforderungen pro Sekunde an symmetrisch verschlüsselte KMS-Schlüssel haben, einschließlich Anfragen, die einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher verwenden.

- Anforderungskontingent für benutzerdefinierte Schlüsselspeicher (pro benutzerdefiniertem Schlüsselspeicher)

  Anforderungen für kryptografische Operationen an KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher werden ebenfalls auf ein `Custom key store request quota` von 1 800 Operationen pro Sekunde. Diese Kontingente werden für jeden benutzerdefinierten Schlüsselspeicher separat berechnet. Sie können Anfragen von mehreren Personen enthalten AWS-Konten , die KMS-Schlüssel im benutzerdefinierten Schlüsselspeicher verwenden.

Wenn Sie beispielsweise eine Verschlüsselungsoperation für einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher in der Region USA Ost (Nord-Virginia) (us-east-1) anfordern, wird diese Anforderung auf das `Cryptographic operations (symmetric) request rate`-Kontingent auf Kontoebene (50 000 Anforderungen pro Sekunde) für ihr Konto und ihre Region, und auf ein `Custom key store request quota` (1 800 Anforderungen pro Sekunde) für ihren benutzerdefinierten Schlüsselspeicher angerechnet. Eine Anforderung für einen Verwaltungsvorgang,

z. B. für einen KMS-Schlüssel in einem benutzerdefinierten Schlüsselspeicher PutKeyPolicy, gilt jedoch nur für das Kontingent auf Kontoebene (15 Anfragen pro Sekunde).

Drosselung AWS KMS von Anfragen

Um sicherzustellen, dass API-Anfragen aller Kunden schnell und zuverlässig beantwortet werden AWS KMS können, werden API-Anfragen, die bestimmte Grenzen überschreiten, gedrosselt.

Eine Drosselung erfolgt, wenn eine Anfrage, die ansonsten gültig sein könnte, AWS KMS zurückgewiesen wird und ein `ThrottlingException` Fehler wie der folgende zurückgegeben wird.

```
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 drosselt Anfragen für die folgenden Bedingungen.

- Die Rate der Anfragen pro Sekunde überschreitet das AWS KMS Anforderungskontingent für ein Konto und eine Region.

  Wenn Benutzer in Ihrem Konto beispielsweise 1000 `DescribeKey` Anfragen in einer Sekunde einreichen, werden alle nachfolgenden `DescribeKey` Anfragen in dieser Sekunde AWS KMS gedrosselt.

  Um auf Drosselung zu reagieren, verwenden Sie eine Backoff- und Wiederholungsstrategie. Diese Strategie wird in einigen AWS SDKs automatisch für HTTP 400-Fehler implementiert.
- Ein Stoß oder eine anhaltende hohe Rate von Anforderungen, um den Status desselben KMS-Schlüssels zu ändern. Diese Bedingung wird oft als „Hotkey“ bezeichnet.

  Wenn beispielsweise eine Anwendung in Ihrem Konto eine permanente Salve von `EnableKey` und `DisableKey` Anfragen nach demselben KMS-Schlüssel sendet, werden die Anfragen AWS KMS gedrosselt. Diese Drosselung erfolgt auch dann, wenn die Anfragen das request-per-second Anforderungslimit für die Operationen und nicht überschreiten. `EnableKey` `DisableKey`

  Um auf Drosselung zu reagieren, passen Sie Ihre Anwendungslogik so an, dass nur erforderliche Anforderungen ausgeführt werden, oder es konsolidiert die Anforderungen mehrerer Funktionen.
- Anfragen für Operationen mit KMS-Schlüsseln in einem AWS CloudHSM Schlüsselspeicher werden möglicherweise mit einer lower-than-expected Geschwindigkeit gedrosselt, wenn der dem

AWS CloudHSM Schlüsselspeicher zugeordnete AWS CloudHSM Cluster zahlreiche Befehle verarbeitet, auch solche, die nichts mit dem Schlüsselspeicher zu tun haben. AWS CloudHSM

(AWS KMS Drosselt Anfragen für Operationen mit KMS-Schlüsseln in einem AWS CloudHSM Schlüsselspeicher nicht mehr, wenn keine PKCS #11 -Sitzungen für den Cluster verfügbar sind. AWS CloudHSM Stattdessen wird eine Meldung ausgelöst `KMSInternalException` und es wird empfohlen, dass Sie Ihre Anfrage erneut versuchen.)

Um Trends in Ihren Anforderungsraten anzuzeigen, verwenden Sie die Service-Quotas-Konsole. Sie können auch einen CloudWatchAmazon-Alarm einrichten, der Sie benachrichtigt, wenn Ihre Anforderungsrate einen bestimmten Prozentsatz eines Kontingents erreicht. Einzelheiten finden Sie im AWS Sicherheitsblog unter Verwalten Sie Ihre AWS KMS API-Anforderungsraten mithilfe von Service Quotas und Amazon CloudWatch.

Alle AWS KMS Kontingente sind anpassbar, mit Ausnahme des Ressourcenkontingents für die Größe des wichtigsten Richtliniendokuments, des Ressourcenkontingents für die On-Demand-Rotation und des Kontingents für AWS CloudHSM wichtige Store-Anfragen. Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch. Um eine Senkung des Kontingents zu beantragen, ein Kontingent zu ändern, das nicht in Service Quotas aufgeführt ist, oder um ein Kontingent AWS-Region in einem Bereich zu ändern, für AWS KMS den Servicekontingente nicht verfügbar sind, besuchen Sie bitte das AWS Support Center und erstellen Sie einen Fall.

> **Note**
>
> AWS KMS benutzerdefinierte Schlüsselspeicher-Anforderungskontingente werden nicht in der Service Quotas Quotas-Konsole angezeigt. Sie können diese Kontingente nicht mithilfe von Service-Quotas-API-Vorgängen anzeigen oder verwalten. Um eine Änderung Ihres Anforderungskontingents für externe Schlüsselspeicher zu beantragen, erstellen Sie im AWS Support Center einen Fall.

Verwendung von AWS KMS durch AWS-Service

Viele AWS-Services unterstützen mithilfe von AWS KMS die Verschlüsselung Ihrer Daten. Wenn ein AWS-Service mit AWS KMS integriert ist, können Sie mithilfe der AWS KMS keys in Ihrem Konto die Daten schützen, die der Service für Sie empfängt, speichert oder verwaltet. Eine vollständige Liste der AWS-Services, die in AWS KMS integriert sind, finden Sie unter AWS-Service-Integration.

Die folgenden Themen erläutern ausführlich, wie bestimmte Services AWS KMS nutzen, einschließlich der von ihnen unterstützten KMS-Schlüssel, wie sie Datenschlüssel verwalten, welche Berechtigungen sie benötigen und wie die Nutzung der KMS-Schlüssel durch die einzelnen Services in Ihrem Konto verfolgt wird.

> **Important**
>
> AWS-Services, die mit AWS KMS integriert sind, verwenden zum Verschlüsseln Ihrer Daten nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Diese Services unterstützen keine Verschlüsselung mit asymmetrischen KMS-Schlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Themen

- So WorkMail verwendet Amazon AWS KMS
- Wie WorkSpaces verwendet AWS KMS

Wie AWS CloudTrailAWS KMS verwendet

Mithilfe von AWS CloudTrail können Sie AWS-API-Aufrufe und andere Aktivitäten in Ihrem AWS-Konto erfassen und die aufgezeichneten Informationen in Protokolldateien in einem Amazon Simple Storage Service (Amazon S3)-Bucket Ihrer Wahl speichern. Standardmäßig werden die Protokolldateien, die in Ihrem S3-Bucket CloudTrail ablegt, mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Alternativ können Sie auch die serverseitige Verschlüsselung mit von verwalteten Schlüsseln (SSE-KMS) wählen. Informationen zum Verschlüsseln Ihrer CloudTrail Protokolldateien mit AWS KMSfinden Sie unter Verschlüsseln von CloudTrail Protokolldateien mit AWS KMS keys (SSE-KMS) im AWS CloudTrail - Benutzerhandbuch.

> **Important**
>
> AWS CloudTrail und Amazon S3 unterstützen nur symmetrische AWS KMS keys. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Ihre CloudTrail Protokolle zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Sie zahlen keine Gebühr für die Schlüsselnutzung, wenn Sie Protokolldateien CloudTrail lesen oder schreiben, die mit einem SSE-KMS-Schlüssel verschlüsselt sind. Sie zahlen jedoch eine Gebühr für die Schlüsselnutzung, wenn Sie auf CloudTrail Protokolldateien zugreifen, die mit einem SSE-KMS-Schlüssel verschlüsselt sind. Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service Pricing (Preise für WAF). Weitere Informationen zu CloudTrail Preisen finden Sie unter -AWS CloudTrailPreise und Verwalten von Kosten im AWS CloudTrail -Benutzerhandbuch.

Themen

- Verstehen, wann Ihr KMS-Schlüssel verwendet wird

Verstehen, wann Ihr KMS-Schlüssel verwendet wird

Das Verschlüsseln von CloudTrail Protokolldateien mit AWS KMS baut auf der Amazon S3-Funktion auf, die als serverseitige Verschlüsselung mit einem AWS KMS key (SSE-KMS) bezeichnet wird. Weitere Informationen zu SSE-KMS finden Sie unter Wie Amazon Simple Storage Service (Amazon S3) AWS KMS nutzt in diesem Handbuch oder unter Schützen von Daten mithilfe der serverseitigen Verschlüsselung mit KMS-Schlüsseln (SSE-KMS) im Amazon-Simple-Storage-Service-Benutzerhandbuch.

Wenn Sie für AWS CloudTrail die Verwendung von SSE-KMS zum Verschlüsseln Ihrer Protokolldateien konfigurieren CloudTrail und Amazon S3 Ihre verwendet, AWS KMS keys wenn Sie bestimmte Aktionen mit diesen Services ausführen. In den folgenden Abschnitten erläutern wir Ihnen, wann und wie diese Services Ihren KMS-Schlüssel verwenden können. Außerdem finden Sie weiterführende Informationen, anhand derer Sie diese Erklärungen praktisch nachvollziehen können.

Aktionen, die dazu führen, dass CloudTrail und Amazon S3 Ihren KMS-Schlüssel verwenden

- Sie konfigurieren CloudTrail , um Protokolldateien mit Ihrem zu verschlüsseln AWS KMS key
- CloudTrail speichert eine Protokolldatei in Ihrem S3-Bucket
- Sie erhalten eine verschlüsselte Protokolldatei aus Ihrem S3-Bucket

Sie konfigurieren CloudTrail , um Protokolldateien mit Ihrem zu verschlüsseln AWS KMS key

Wenn Sie Ihre CloudTrail Konfiguration aktualisieren, um Ihren KMS-Schlüssel zu verwenden, CloudTrail sendet eine `GenerateDataKey` Anforderung an , AWS KMS um zu überprüfen, ob der KMS-Schlüssel vorhanden ist und die Berechtigung CloudTrail hat, ihn für die Verschlüsselung zu verwenden. verwendet nicht den CloudTrail resultierenden Datenschlüssel.

Die Anforderung des Typs `GenerateDataKey` enthält die folgenden Informationen für denVerschlüsselungskontext:

- Der Amazon-Ressourcenname (ARN) des CloudTrail Trails
- Der ARN des S3-Buckets und des Pfads, an den die CloudTrail Protokolldateien übermittelt werden

Die `GenerateDataKey` Anforderung führt zu einem Eintrag in Ihren CloudTrail Protokollen, der dem folgenden Beispiel ähnelt. Wenn Sie einen solchen Protokolleintrag sehen, können Sie feststellen, dass CloudTrail

(1))
die AWS KMS
(2))-Gene
(3))
für einen bestimmten Trail () aufgerufen
hat4 .
hat den Datenschlüssel unter einem bestimmten KMS-Schlüssel () AWS KMS
erstellt5

> **Note**
>
> Möglicherweise müssen Sie nach rechts scrollen, um einige der Textfelder im folgenden Beispielprotokolleintrag anzuzeigen.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::086441151436:user/
AWSCloudTrail",1
    "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",2
  "eventName":
 "GenerateDataKey",3
  "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",4
      "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",5
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

CloudTrail speichert eine Protokolldatei in Ihrem S3-Bucket

Jedes Mal, wenn eine Protokolldatei in Ihren S3-Bucket CloudTrail einfügt, sendet Amazon S3 AWS KMS im Namen von eine `GenerateDataKey` Anforderung an CloudTrail. Als Antwort auf diese Anforderung generiert AWS KMS einen eindeutigen Datenschlüssel und sendet zwei Kopien dieses Datenschlüssels an Amazon S3: eine Kopie als Klartext und eine Kopie, die mit dem angegebenen KMS-Schlüssel verschlüsselt ist. Amazon S3 verwendet den Klartext-Datenschlüssel, um die CloudTrail Protokolldatei zu verschlüsseln, und entfernt dann den Klartext-Datenschlüssel so schnell wie möglich nach der Verwendung aus dem Speicher. Amazon S3 speichert den verschlüsselten Datenschlüssel als Metadaten mit der verschlüsselten CloudTrail Protokolldatei.

Die Anforderung des Typs `GenerateDataKey` enthält die folgenden Informationen für denVerschlüsselungskontext:

- Der Amazon-Ressourcenname (ARN) des CloudTrail Trails
- Der ARN des S3-Objekts (die CloudTrail Protokolldatei)

Jede `GenerateDataKey` Anforderung führt zu einem Eintrag in Ihren CloudTrail Protokollen, der dem folgenden Beispiel ähnelt. Wenn Sie einen solchen Protokolleintrag sehen, können Sie feststellen, dass CloudTrail (1) die AWS KMS (2)-Gene (3) für einen bestimmten Trail () aufgerufen hat, um eine bestimmte Protokolldatei (4) zu schützen5 hat den Datenschlüssel unter dem angegebenen KMS-Schlüssel () AWS KMS erstellt6 der zweimal im selben Protokolleintrag angezeigt wird.

> **Note**
>
> Möglicherweise müssen Sie nach rechts scrollen, um einige der Textfelder im folgenden Beispielprotokolleintrag anzuzeigen.

```
{
  "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"
}
```

Sie erhalten eine verschlüsselte Protokolldatei aus Ihrem S3-Bucket

Jedes Mal, wenn Sie eine verschlüsselte CloudTrail Protokolldatei aus Ihrem S3-Bucket erhalten, sendet Amazon S3 AWS KMS in Ihrem Namen eine `Decrypt` Anforderung an , um den verschlüsselten Datenschlüssel der Protokolldatei zu entschlüsseln. Als Antwort auf diese Anforderung entschlüsselt AWS KMS mithilfe Ihres KMS-Schlüssels den Datenschlüssel und sendet diesen dann als Klartext-Datenschlüssel an Amazon S3. Amazon S3 verwendet den Klartext-Datenschlüssel, um die CloudTrail Protokolldatei zu entschlüsseln, und entfernt dann den Klartext-Datenschlüssel so schnell wie möglich nach der Verwendung aus dem Speicher.

Die Anforderung des Typs `Decrypt` enthält die folgenden Informationen für denVerschlüsselungskontext:

- Der Amazon-Ressourcenname (ARN) des CloudTrail Trails
- Der ARN des S3-Objekts (die CloudTrail Protokolldatei)

Jede `Decrypt` Anforderung führt zu einem Eintrag in Ihren CloudTrail Protokollen, der dem folgenden Beispiel ähnelt. Einem solchen Protokolleintrag können Sie entnehmen, dass ein Benutzer in Ihrem AWS-Konto
(1)
die AWS KMS
(2)
`Decrypt`-Operation
(3)
für einen bestimmten Trail
(4)
und eine bestimmte Protokolldatei
(5)
aufgerufen hat. AWS KMS hat den Datenschlüssel mit einem bestimmten KMS-Schlüssel
(6)
entschlüsselt.

> **Note**
>
> Möglicherweise müssen Sie nach rechts scrollen, um einige der Textfelder im folgenden Beispielprotokolleintrag anzuzeigen.

```
{
  "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"
}
```

So verwendet Amazon DynamoDB AWS KMS

Amazon DynamoDB ist ein vollständig verwalteter, skalierbarer NoSQL-Datenbankservice. DynamoDB ist mit AWS Key Management Service (AWS KMS) integriert, um die serverseitige Verschlüsselungsfunktion Verschlüsselung im Ruhezustand zu unterstützen.

Mit Verschlüsselung im Ruhezustand verschlüsselt DynamoDB alle Kundendaten transparent in einer DynamoDB-Tabelle, einschließlich des Primärschlüssels und der lokalen und globalen sekundären Indizes, wenn die Tabelle dauerhaft auf dem Datenträger verbleibt. (Wenn Ihre Tabelle einen Sortierschlüssel hat, werden einige der Sortierschlüssel, die Bereichsgrenzen markieren, in Klartext in den Metadaten der Tabelle gespeichert.) Wenn Sie auf Ihre Tabelle zugreifen, entschlüsselt DynamoDB die Tabellendaten transparent. Sie müssen Anwendungen nicht ändern, um verschlüsselte Tabellen verwenden oder verwalten zu können.

Die Verschlüsselung im Ruhezustand schützt außerdem DynamoDB-Streams, globale Tabellen und Backups, jedes Mal, wenn diese Objekte auf beständigen Medien gespeichert werden. Anweisungen über Tabellen in diesem Thema gelten auch für diese Objekte.

Alle DynamoDB-Tabellen werden verschlüsselt. Es gibt keine Option zum Aktivieren oder Deaktivieren der Verschlüsselung für neue oder bestehende Tabellen. Standardmäßig werden alle Tabellen mit einem AWS-eigener Schlüssel im DynamoDB-Servicekonto verschlüsselt. Sie können jedoch eine Option zum Verschlüsseln einiger oder aller Ihrer Tabellen mit einem kundenverwalteten Schlüssel oder dem Von AWS verwalteter Schlüssel für DynamoDB in Ihrem Konto auswählen.

Einzelheiten zur Amazon-DynamoDB-Unterstützung für KMS-Schlüssel finden Sie unter Ruhende DynamoDB-Verschlüsselung im Entwicklerhandbuch von Amazon DynamoDB.

Wie Amazon Elastic Block Store (Amazon EBS) AWS KMS nutzt.

In diesem Thema wird im Detail erläutert, wie Amazon Elastic Block Store (Amazon EBS) AWS KMS nutzt, um Volumes und Snapshots zu verschlüsseln. Grundlegende Anweisungen zur Verschlüsselung von EBS-Volumes finden Sie unter Amazon EBS-Verschlüsselung.

Themen

Amazon-EBS-Verschlüsselung

Wenn Sie ein verschlüsseltes Amazon-EBS-Volume einem unterstützten Amazon Elastic Compute Cloud (Amazon EC2)-Instance-Typ zuordnen, werden sowohl die auf dem Volume gespeicherten Daten als auch die Datenträger-E/A und die von dem Volume erstellten Snapshots verschlüsselt. Die Verschlüsselung erfolgt auf den Servern, die Amazon-EC2-Instances hosten.

Diese Funktion wird für alle Typen von Amazon-EBS-Volumes unterstützt. Der Zugriff auf verschlüsselte Volumes erfolgt genau wie der Zugriff auf andere Volumes. Die Ver- und Entschlüsselung werden transparent gehandhabt und erfordern keine weitere Aktion von Ihnen, Ihrer EC2-Instance oder Ihrer Anwendung. Snapshots von verschlüsselten Volumes werden automatisch verschlüsselt, genau wie Volumes, die auf Basis verschlüsselter Snapshots erstellt werden.

Der Verschlüsselungsstatus eines EBS-Volume wird bei der Erstellung des Volume bestimmt. Der Verschlüsselungsstatus eines vorhandenen Volume kann nicht geändert werden. Sie können jedoch Daten zwischen verschlüsselten und nicht verschlüsselten Volumes migrieren und beim Kopieren eines Snapshots einen neuen Verschlüsselungsstatus anwenden.

Amazon EBS unterstützt standardmäßig optionale Verschlüsselung. Sie können die Verschlüsselung automatisch auf allen neuen EBS-Volumes und Snapshot-Kopien in Ihrem AWS-Konto und Ihrer Region aktivieren. Diese Konfigurationseinstellung hat keine Auswirkungen auf vorhandene Volumes oder Snapshots. Details dazu finden Sie unter Standardmäßige Verschlüsselung im Amazon-EC2-Benutzerhandbuch für Linux-Instances oder Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Verwenden von KMS-Schlüsseln und Datenschlüsseln

Wenn Sie ein verschlüsseltes Amazon-EBS-Volume erstellen, geben Sie einen AWS KMS key an. Amazon EBS verwendet standardmäßig Von AWS verwalteter Schlüssel für Amazon EBS in Ihrem Konto (`aws/ebs`). Sie können jedoch einen kundenverwalteten Schlüssel angeben, den Sie erstellen und verwalten.

Um einen kundenverwalteten Schlüssel zu verwenden, müssen Sie Amazon EBS die Berechtigung zur Verwendung des KMS-Schlüssels in Ihrem Namen erteilen. Eine Liste der erforderlichen Berechtigungen finden Sie unter Berechtigungen für IAM-Benutzer im Amazon-EC2-Benutzerhandbuch für Linux-Instances oder Amazon-EC2 Benutzerhandbuch für Windows-Instances.

> **Important**
>
> Amazon EBS unterstützt nur symmetrische KMS-Schlüssel. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um ein Amazon-EBS-Volume zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Amazon EBS holt sich für jedes Volume einen von AWS KMS erstellten eindeutigen Datenschlüssel, der mit dem von Ihnen angegebenen KMS-Schlüssel verschlüsselt ist. Amazon EBS speichert den verschlüsselten Datenschlüssel mit dem Volume. Wenn Sie das Volume dann an eine Amazon-EC2-Instance anhängen, ruft Amazon EBS AWS KMS auf, um den Datenschlüssel zu entschlüsseln. Amazon EBS verwendet den Klartext-Datenschlüssel im Hypervisor-Speicher, um den Disk-I/O zum Volume zu verschlüsseln. Details dazu finden Sie unter Wie EBS-Verschlüsselung funktioniert im Amazon-EC2-Benutzerhandbuch für Linux-Instances oder Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Amazon-EBS-Verschlüsselungskontext

In seinen - GenerateDataKeyWithoutPlaintext und Decrypt-Anforderungen an verwendet AWS KMSAmazon EBS einen Verschlüsselungskontext mit einem Name-Wert-Paar, das das Volume oder den Snapshot in der Anforderung identifiziert. Der Name im Verschlüsselungskontext variiert nicht.

Ein Verschlüsselungskontext ist eine Gruppe von Schlüssel/Wert-Paaren mit willkürlichen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext kryptografisch an die verschlüsselten Daten. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Für alle Volumes und für verschlüsselte Snapshots, die mit der Amazon-EBS-CreateSnapshotOperation erstellt wurden, verwendet Amazon EBS die Volume-ID als Verschlüsselungskontextwert. Im `requestParameters`-Feld eines CloudTrail-Protokolleintrags sieht der Verschlüsselungskontext wie folgt aus:

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Für verschlüsselte Snapshots, die mit der Amazon EC2-CopySnapshotOperation erstellt wurden, verwendet Amazon EBS die Snapshot-ID als Verschlüsselungskontextwert. Im `requestParameters`-Feld eines CloudTrail-Protokolleintrags sieht der Verschlüsselungskontext wie folgt aus:

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Erkennen von Amazon-EBS-Fehlern

Um ein verschlüsseltes EBS-Volume zu erstellen oder das Volume an eine EC2-Instance anzufügen, müssen Amazon EBS und die Amazon-EC2-Infrastruktur in der Lage sein, den für die EBS-Volume-Verschlüsselung angegebenen KMS-Schlüssel zu verwenden. Wenn der KMS-Schlüssel nicht verwendbar ist, z. B. wenn sein Schlüsselstatus nicht `Enabled` ist, schlägt die Volume-Erstellung oder das Anfügen des Volumes fehl.

In diesem Fall sendet Amazon EBS ein Ereignis an Amazon EventBridge (früher CloudWatch Ereignisse), um Sie über den Fehler zu informieren. In können Sie Regeln einrichten EventBridge, die als Reaktion auf diese Ereignisse automatische Aktionen auslösen. Weitere Informationen finden Sie unter Amazon CloudWatch Events für Amazon EBS im Amazon EC2-Benutzerhandbuch für Linux-Instances, insbesondere in den folgenden Abschnitten:

- Ungültiger Verschlüsselungsschlüssel für das Anfügen oder erneute Anfügen eines Volume
- Ungültiger Verschlüsselungsschlüssel für das Erstellen eines Volume

Um dieser Fehler zu beheben, vergewissern Sie sich, dass der KMS-Schlüssel, den Sie für die EBS-Volume-Verschlüsselung angegeben haben, aktiviert ist. Zu diesem Zweck zeigen Sie zunächst den KMS-Schlüssel an, um seinen aktuellen Schlüsselstatus festzustellen (die Status-Spalte in der Konsole). Sehen Sie sich dann die Informationen unter einem der folgenden Links an:

- Wenn der Schlüsselstatus des KMS-Schlüssels deaktiviert ist, aktivieren Sie ihn.
- Wenn der Schlüsselstatus des KMS-Schlüssels zeigt, dass der Import ausstehend ist, importieren Sie das Schlüsselmaterial.

- Wenn der Schlüsselstatus des KMS-Schlüssels Löschung ausstehend ist, brechen Sie die Schlüssellöschung ab.

Verwenden von AWS CloudFormation zum Erstellen von verschlüsselten Amazon-EBS-Volumes

Sie können AWS CloudFormation verwenden, um verschlüsselte Amazon-EBS-Volumes zu erstellen. Weitere Informationen finden Sie unter AWS::EC2::Volume im AWS CloudFormation-Benutzerhandbuch.

Wie Amazon Elastic Transcoder AWS KMS nutzt

Mit Amazon Elastic Transcoder können Sie in einem Amazon-S3-Bucket gespeicherte Mediendateien in Formate konvertieren, die von Wiedergabegeräten aus dem Verbrauchersegment abgespielt werden können. Sowohl die Eingabe- als auch die Ausgabedateien lassen sich verschlüsseln und entschlüsseln. In den folgenden Abschnitten wird beschrieben, wie AWS KMS für diese beiden Prozesse verwendet wird.

Themen

Verschlüsseln der Eingabedatei

Bevor Sie Elastic Transcoder verwenden können, müssen Sie einen Amazon-S3-Bucket erstellen und Ihre Mediendatei in diesen Bucket hochladen. Sie können die Datei entweder vor dem Upload mithilfe clientseitiger AES-Verschlüsselung verschlüsseln oder nach dem Upload mithilfe der serverseitigen Verschlüsselung von Amazon S3.

Wenn Sie sich für die clientseitige Verschlüsselung mit AES entscheiden, müssen Sie die Datei vor dem Upload in Amazon S3 selbst verschlüsseln und Elastic Transcoder Zugriff auf den Verschlüsselungsschlüssel geben. Hierzu verwenden Sie einen symmetrischen AWS KMS

AWS KMS key zum Schutz des AES-Verschlüsselungsschlüssels, mit dem Sie die Mediendatei verschlüsselt haben.

Wenn Sie sich für die serverseitige Verschlüsselung entscheiden, erlauben Sie Amazon S3, sämtliche Verschlüsselungs- und Entschlüsselungsoperationen in Ihrem Namen auf die Dateien anzuwenden. Bei der Konfiguration von Amazon S3 haben Sie die Wahl zwischen drei unterschiedlichen Verschlüsselungsschlüsseln, mit denen der zur Verschlüsselung Ihrer Datei verwendete eindeutige Datenschlüssel verschlüsselt werden kann:

- Ein Amazon-S3-Schlüssel, ein Verschlüsselungsschlüssel, den Amazon S3 besitzt und verwaltet. Er ist nicht Teil Ihres AWS-Konto.
- Der Von AWS verwalteter Schlüssel für Amazon S3, ein KMS-Schlüssel, der Teil Ihres Kontos ist, aber von AWS verwaltet wird
- Ein beliebiger symmetrischer kundenverwalteter Schlüssel, den Sie mit AWS KMS erstellen.

> **Important**
>
> Für die clientseitige und serverseitige Verschlüsselung unterstützt Elastic Transcoder nur symmetrische KMS-Schlüssel. Sie können keine asymmetrische KMS-Schlüssel verwenden, um Ihre Elastic-Transcoder-Dateien zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Sie können die Verschlüsselung aktivieren und einen Schlüssel mithilfe der Amazon-S3-Konsole oder der entsprechenden Amazon-S3-APIs angeben. Weitere Informationen zur Verschlüsselung in Amazon S3 finden Sie unter Schützen von Daten mithilfe der serverseitigen Verschlüsselung mit KMS-Schlüsseln (SSE-KMS) im Amazon-Simple-Storage-Service-Benutzerhandbuch.

Wenn Sie die Eingabedatei mithilfe des Von AWS verwalteter Schlüssel für Amazon S3 in Ihrem Konto oder mithilfe eines kundenverwalteten Schlüssels schützen, interagieren Amazon S3 und AWS KMS wie folgt:

1. Amazon S3 fordert einen Klartext-Datenschlüssel und eine mit dem angegebenen KMS-Schlüssel verschlüsselte Kopie dieses Datenschlüssels an.

2. AWS KMS erstellt einen Datenschlüssel, verschlüsselt diesen Datenschlüssel mit dem angegebenen KMS-Schlüssel und sendet sowohl den Klartext-Datenschlüssel als auch den verschlüsselten Datenschlüssel an Amazon S3.
3. Amazon S3 verschlüsselt die Mediendatei mit dem Klartext-Datenschlüssel und speichert die Datei anschließend im angegebenen Amazon-S3-Bucket.
4. Amazon S3 speichert den verschlüsselten Datenschlüssel zusammen mit der verschlüsselten Mediendatei.

Entschlüsseln der Eingabedatei

Wenn Sie die Eingabedatei mithilfe der serverseitigen Verschlüsselung von Amazon S3 verschlüsseln, entschlüsselt Elastic Transcoder die Datei nicht. Stattdessen nutzt Elastic Transcoder zur Entschlüsselung Amazon S3, und zwar gemäß den Einstellungen, die Sie bei der Erstellung des Auftrags und der Pipeline festgelegt haben.

Möglich sind die nachfolgend aufgeführten Einstellungskombinationen.

Verschlüsselungsmodus	AWS KMS-Schlüssel	Bedeutung
`S3`	Standard	Amazon S3 erstellt und verwaltet die Schlüssel, mit denen die Mediendatei verschlüsselt und entschlüsselt wird. Dieser Prozess ist für den Benutzer nicht transparent.
`S3-AWS-KMS`	Standard	Amazon S3 verschlüsselt die Mediendatei mit einem Datenschlüssel, der mit dem standardmäßigen Von AWS verwalteter Schlüssel für Amazon S3 in Ihrem Konto verschlüsselt wurde.
`S3-AWS-KMS`	Benutzerdefiniert (mit ARN)	Amazon S3 benutzt ein Datenschlüssel, der mit dem

Verschlüsselungsmodus	AWS KMS-Schlüssel	Bedeutung
		angegebenen kundenverwalteten KMS-Schlüssel verschlüsselt wurde, um die Mediendatei mit einem Datenschlüssel zu verschlüsseln.

Wenn Sie `S3-AWS-KMS` angeben, interagieren Amazon S3 und AWS KMS bei der Entschlüsselung wie folgt:

1. Amazon S3 sendet den verschlüsselten Datenschlüssel an AWS KMS.
2. AWS KMS entschlüsselt den Datenschlüssel mithilfe des passenden KMS-Schlüssels und sendet anschließend den Klartext-Datenschlüssel an Amazon S3 zurück.
3. Amazon S3 entschlüsselt den Chiffretext mithilfe des Klartext-Datenschlüssels.

Wenn Sie sich für die clientseitige Verschlüsselung mit einem AES-Schlüssel entscheiden, ruft Elastic Transcoder die verschlüsselte Datei aus dem Amazon-S3-Bucket ab und entschlüsselt sie. Zur Entschlüsselung des AES-Schlüssels verwendet Elastic Transcoder den KMS-Schlüssel, den Sie bei der Erstellung der Pipeline angegeben haben. Anschließend wird die Mediendatei mithilfe dieses AES-Schlüssels entschlüsselt.

Verschlüsseln der Ausgabedatei

Elastic Transcoder verschlüsselt die Ausgabedatei gemäß den Verschlüsselungs-Einstellungen, die Sie bei der Erstellung des Auftrags und der Pipeline angegeben haben. Verfügbar sind die nachfolgend aufgeführten Optionen.

Verschlüsselungsmodus	AWS KMS-Schlüssel	Bedeutung
`S3`	Standard	Amazon S3 erstellt und verwaltet die Schlüssel, mit denen die Ausgabedatei verschlüsselt wird.

Verschlüsselungsmodus	AWS KMS-Schlüssel	Bedeutung
S3-AWS-KMS	Standard	Amazon S3 verwendet einen Datenschlüssel, der von AWS KMS erstellt und mit dem Von AWS verwalteter Schlüssel für Amazon S3 in Ihrem Konto verschlüsselt wird.
S3-AWS-KMS	Benutzerdefiniert (mit ARN)	Amazon S3 verschlüsselt die Mediendatei mit einem Datenschlüssel, der mit dem durch den ARN angegebenen kundenverwalteten KMS-Schlüssel verschlüsselt wurde.
AES-	Standard	Elastic Transcoder entschlüsselt den von Ihnen angegebenen AES-Schlüssel mithilfe des Von AWS verwalteter Schlüssel für Amazon S3 in Ihrem Konto und verschlüsselt die Ausgabedatei anschließend mit diesem Schlüssel.
AES-	Benutzerdefiniert (mit ARN)	Elastic Transcoder entschlüsselt den von Ihnen angegebenen AES-Schlüssel mithilfe des durch den ARN angegebenen kundenverwalteten KMS-Schlüssel und verschlüsselt die Ausgabedatei anschließend mit diesem Schlüssel.

Wenn Sie festlegen, dass die Ausgabedatei mit einem Von AWS verwalteter Schlüssel für Amazon S3 in Ihrem Konto oder mit einem kundenverwalteten KMS-Schlüssel verschlüsselt wird, interagieren Amazon S3 und AWS KMS wie folgt:

1. Amazon S3 fordert einen Klartext-Datenschlüssel und eine mit dem angegebenen KMS-Schlüssel verschlüsselte Kopie dieses Datenschlüssels an.
2. AWS KMS erstellt einen Datenschlüssel, verschlüsselt diesen Datenschlüssel mit dem KMS-Schlüssel und sendet sowohl den Klartext-Datenschlüssel als auch den verschlüsselten Datenschlüssel an Amazon S3.
3. Amazon S3 verschlüsselt die Mediendatei mit dem Datenschlüssel und speichert sie im angegebenen Amazon-S3-Bucket.
4. Amazon S3 speichert den verschlüsselten Datenschlüssel zusammen mit der verschlüsselten Mediendatei.

Wenn die Ausgabedatei mit dem von Ihnen angegebenen AES-Schlüssel verschlüsselt werden soll, muss dieser AES-Schlüssel mit einem KMS-Schlüssel in AWS KMS verschlüsselt werden. Elastic Transcoder, AWS KMS und Sie interagieren dann wie folgt:

1. Sie verschlüsseln Ihren AES-Schlüssel durch Aufrufen der Encrypt-Operation in der AWS KMS-API. AWS KMS verschlüsselt den Schlüssel dann mit dem angegebenen KMS-Schlüssel. Welcher KMS-Schlüssel verwendet werden soll, geben Sie bei der Erstellung der Pipeline an.
2. Sie geben bei der Erstellung des Elastic-Transcoder-Auftrags die Datei an, die den verschlüsselten AES-Schlüssel enthält.
3. Elastic Transcoder entschlüsselt den Schlüssel durch Aufrufen der Decrypt-Operation in der AWS KMS-API. Dabei wird der verschlüsselte Schlüssel als Chiffretext übergeben.
4. Elastic Transcoder verschlüsselt die Ausgabemediendatei mit dem entschlüsselten AES-Schlüssel und löscht den entschlüsselten AES-Schlüssel dann aus dem Arbeitsspeicher. Auf dem Datenträger wird nur die verschlüsselte Kopie gespeichert, die Sie ursprünglich im Job angegeben haben.
5. Nun können Sie die verschlüsselte Ausgabedatei herunterladen und lokal mithilfe des ursprünglich von Ihnen angegebenen AES-Schlüssels entschlüsseln.

> **Important**
>
> AWS speichert niemals Ihre privaten Verschlüsselungsschlüssel. Daher ist es wichtig, dass Sie Ihre Schlüssel sicher und geschützt verwalten. Wenn die Schlüssel verloren gehen, können Sie Ihre Daten nicht mehr entschlüsseln.

Schützen von HLS-Inhalten

HTTP Live Streaming (HLS) ist ein adaptives Streaming-Protokoll. Elastic Transcoder unterstützt HLS, indem Ihre Eingabedatei in kleinere Einzeldateien, genannt Mediensegmente, aufgesplittet wird. Ein Satz aus entsprechenden einzelnen Mediensegmenten enthält dasselbe Material jeweils mit einer anderen Bitrate kodiert. Das Wiedergabegerät kann so den Stream auswählen, der am besten für die verfügbare Bandbreite geeignet ist. Elastic Transcoder erstellt außerdem Wiedergabelisten, die Metadaten zu den verschiedenen zum Streaming verfügbaren Segmenten enthalten.

Wenn Sie den Schutz von HLS-Inhalten aktivieren, wird jedes Mediensegment mit einem 128-Bit-AES-Verschlüsselungsschlüssel verschlüsselt. Wenn die Inhalte abgespielt werden, lädt das Wiedergabegerät während der Wiedergabe den Schlüssel herunter und entschlüsselt die Mediensegmente.

Es werden zwei Typen von Schlüsseln verwendet: ein KMS-Schlüssel und ein Datenschlüssel. Sie müssen einen KMS-Schlüssel für die Verschlüsselung und Entschlüsselung des Datenschlüssels erstellen. Elastic Transcoder nutzt den Datenschlüssel zum Verschlüsseln und Entschlüsseln der Mediensegmente. Der Datenschlüssel muss ein AES-128-Schlüssel sein. Alle Varianten und Segmente ein und desselben Inhalts werden mit demselben Datenschlüssel verschlüsselt. Sie können selbst einen Datenschlüssel angeben oder Elastic Transcoder einen Datenschlüssel erstellen lassen.

Mit dem KMS-Schlüssel kann der Datenschlüssel zu den folgenden Zeitpunkten verschlüsselt werden:

- Wenn Sie einen eigenen Datenschlüssel bereitstellen, müssen Sie diesen Schlüssel verschlüsseln, bevor Sie ihn an Elastic Transcoder übergeben.
- Wenn Sie einen Datenschlüssel von Elastic Transcoder anfordern, verschlüsselt Elastic Transcoder den Datenschlüssel für Sie.

Mit dem KMS-Schlüssel kann der Datenschlüssel zu folgenden Zeitpunkten entschlüsselt werden:

- Elastic Transcoder entschlüsselt den von Ihnen angegebenen Datenschlüssel, wenn mit diesem Datenschlüssel die Ausgabedatei verschlüsselt oder die Eingabedatei entschlüsselt werden muss.
- Sie entschlüsseln einen von Elastic Transcoder generierten Datenschlüssel, um damit anschließend Ausgabedateien zu entschlüsseln.

Weitere Informationen finden Sie unter HLS Inhaltsschutz im Entwicklerhandbuch für Amazon Elastic Transcoder.

Elastic-Transcoder-Verschlüsselungskontext

Ein Verschlüsselungskontext ist eine Gruppe von Schlüssel/Wert-Paaren mit willkürlichen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext kryptografisch an die verschlüsselten Daten. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Elastic Transcoder verwendet denselben Verschlüsselungskontext in allen AWS KMS-API-Anforderungen, um Datenschlüssel zu generieren, zu verschlüsseln und zu entschlüsseln.

```
"service" : "elastictranscoder.amazonaws.com"
```

Der Verschlüsselungskontext wird in - CloudTrail Protokolle geschrieben, um Ihnen zu helfen zu verstehen, wie ein bestimmter AWS KMS KMS-Schlüssel verwendet wurde. Im `requestParameters` Feld einer CloudTrail Protokolldatei sieht der Verschlüsselungskontext wie folgt aus:

```
"encryptionContext": {
  "service" : "elastictranscoder.amazonaws.com"
}
```

Weitere Informationen dazu, wie Sie eine der unterstützten Verschlüsselungsoptionen für Elastic-Transcoder-Aufträge konfigurieren können, finden Sie unter Verschlüsselungsoptionen für Daten im Entwicklerhandbuch für Amazon Elastic Transcoder.

Wie Amazon EMR AWS KMS nutzt

Wenn Sie einen Amazon-EMR-Cluster verwenden, können Sie den Cluster zum Verschlüsseln von Daten im Ruhezustand konfigurieren, bevor Sie sie in einem persistenten Speicher ablegen. Sie können ruhende Daten im EMR-Dateisystem (EMRFS) und/oder auf den Speicher-Volumes von Cluster-Knoten verschlüsseln. Sie können Data-at-Rest mit einem AWS KMS key verschlüsseln. In den folgenden Themen wird erläutert, wie ein Amazon-EMR-Cluster einen KMS-Schlüssel zum Verschlüsseln von Daten im Ruhezustand verwendet.

> **Important**
>
> Amazon EMR unterstützt nur symmetrische KMS-Schlüssel. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Data-at-Rest in einem Amazon-EMR-Cluster zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Amazon-EMR-Cluster verschlüsseln auch Daten während der Übertragung, was bedeutet, dass der Cluster Daten vor der Übertragung über das Netzwerk verschlüsselt. Sie können einen KMS-Schlüssel verwenden, um Daten während der Übertragung zu verschlüsseln. Weitere Information erhalten Sie unter Verschlüsselung von Daten während der Übertragung im Amazon-EMR-Management-Handbuch.

Weitere Informationen über alle in Amazon EMR verfügbaren Verschlüsselungsoptionen finden Sie unter Verschlüsselungsoptionen im Amazon-EMR-Management-Handbuch.

Themen

- Verschlüsseln von Daten auf dem EMR-Dateisystem (EMRFS)
- Verschlüsseln von Daten auf den Speicher-Volumes von Cluster-Knoten
- Verschlüsselungskontext

Verschlüsseln von Daten auf dem EMR-Dateisystem (EMRFS)

Amazon-EMR-Cluster verwenden zwei verteilte Dateisysteme:

- Das Hadoop Distributed File System (HDFS). Die HDFS-Verschlüsselung verwendet keinen KMS-Schlüssel in AWS KMS.
- Das EMR File System (EMRFS). Das EMRFS ist eine Implementierung von HDFS, das es Amazon-EMR-Clustern ermöglicht, Daten in Amazon Simple Storage Service (Amazon S3) zu speichern. EMRFS unterstützt vier Verschlüsselungsoptionen, wovon zwei einen KMS-Schlüssel in AWS KMS verwenden. Weitere Informationen über alle in Amazon EMR verfügbaren Verschlüsselungsoptionen finden Sie unter Verschlüsselungsoptionen im Amazon-EMR-Management-Handbuch.

Die zwei EMRFS-Verschlüsselungsoptionen, die einen KMS-Schlüssel verwenden, um die folgenden von Amazon S3 angebotenen Verschlüsselungsfunktionen zu nutzen:

- Schützen von Daten durch serverseitige Verschlüsselung mit AWS Key Management Service (SSE-KMS). Der Amazon EMR-Cluster sendet Daten an Amazon S3. Amazon S3 verwendet einen KMS-Schlüssel, um die Daten vor dem Speichern in einem S3-Bucket zu verschlüsseln. Weitere Information dazu finden Sie unter Prozess für das Verschlüsseln von Daten auf EMRFS mit SSE-KMS.
- Schützen von Daten mithilfe clientseitiger Verschlüsselung (CSE-KMS) Daten in einem Amazon EMR werden unter einemAWS KMS keybevor es zur Speicherung an Amazon S3 gesendet wird. Weitere Information dazu finden Sie unter Prozess für das Verschlüsseln von Daten auf EMRFS mit CSE-KMS.

Wenn Sie einen Amazon-EMR-Cluster konfigurieren, um Daten auf EMRFS mit SSE-KMS oder CSE-KMS zu verschlüsseln, wählen Sie den KMS-Schlüssel in , den Amazon S3 oder der Amazon-EMR-Cluster verwenden soll. Für SSE-KMS können Sie den Von AWS verwalteter Schlüssel für Amazon S3 mit dem Alias aws/s3 oder einen von Ihnen erstellten symmetrischen kundenverwalteten Schlüssel auswählen. Für CSE-KMS müssen Sie einen von Ihnen erstellten symmetrischen kundenverwalteten Schlüssel auswählen. Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie sicherstellen, dass der Amazon-EMR-Cluster über die Berechtigung zur Nutzung des KMS-Schlüssels verfügt. Weitere Information erhalten Sie unter Verschlüsselung von Daten mit AWS KMS keys im Amazon-EMR-Management-Handbuch.

Für beide, SSE-KMS und CSE-KMS, ist der von Ihnen gewählte KMS-Schlüssel der Stammschlüssel in einem Envelope-Verschlüsselungs-Workflow. Die Daten werden mit einem eindeutigenDatenschlüsseldas ist unter dem KMS-Schlüssel verschlüsseltAWS KMSaus. Die verschlüsselten Daten und eine verschlüsselte Kopie des Datenschlüssels werden zusammen als einzelnes verschlüsseltes Objekt in einem S3-Bucket verschlüsselt. Weitere Informationen dazu finden Sie in den folgenden Themen.

Themen

- Prozess für das Verschlüsseln von Daten auf EMRFS mit SSE-KMS
- Prozess für das Verschlüsseln von Daten auf EMRFS mit CSE-KMS

Prozess für das Verschlüsseln von Daten auf EMRFS mit SSE-KMS

Wenn Sie für einen Amazon-EMR-Cluster für die Verwendung von SSE-KMS konfigurieren, läuft der Verschlüsselungsprozess folgendermaßen ab:

1. Der Cluster sendet Daten zu Amazon S3 zur Speicherung in einem S3-Bucket.
2. Amazon S3 sendet eine GenerateDataKey Anforderung an AWS KMS, in der die Schlüssel-ID des KMS-Schlüssels angegeben wird, den Sie bei der Konfiguration des Clusters für die Verwendung von SSE-KMS ausgewählt haben. Die Anforderung enthält Verschlüsselungskontext; weitere Informationen finden Sie unter Verschlüsselungskontext.
3. AWS KMS generiert einen eindeutigen Datenverschlüsselungsschlüssel (Datenschlüssel) und sendet dann zwei Kopien dieses Datenschlüssels an Amazon S3. Eine Kopie ist unverschlüsselt (Klartext) und die andere Kopie ist mit dem KMS-Schlüssel verschlüsselt.
4. Amazon S3 verschlüsselt die in Schritt 1 erhaltenen Daten mit dem Klartext-Datenschlüssel und entfernt diesen Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.
5. Amazon S3 speichert die verschlüsselten Daten und eine verschlüsselte Kopie des Datenschlüssels zusammen als einzelnes verschlüsseltes Objekt in einem S3-Bucket.

Die Entschlüsselung funktioniert wie folgt:

1. Der Cluster fordert ein verschlüsseltes Datenobjekt von einem S3-Bucket an.
2. Amazon S3 extrahiert den verschlüsselten Datenschlüssel aus dem S3-Objekt und sendet dann den verschlüsselten Datenschlüssel an AWS KMS mit einer Decrypt-Anforderung. Die Anforderung enthält einen Verschlüsselungskontext.
3. AWS KMS entschlüsselt den verschlüsselten Datenschlüssel mit dem gleichen KMS-Schlüssel, der für die Verschlüsselung verwendet wurde, und sendet dann den entschlüsselten (Klartext-) Datenschlüssel an Amazon S3.
4. Amazon S3 verwendet den Klartext-Datenschlüssel, um die verschlüsselten Daten zu entschlüsseln, und entfernt den Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.
5. Amazon S3 sendet die entschlüsselten Daten an den Cluster.

Prozess für das Verschlüsseln von Daten auf EMRFS mit CSE-KMS

Wenn Sie für einen Amazon-EMR-Cluster die Verwendung von CSE-KMS konfigurieren, läuft der Verschlüsselungsprozess folgendermaßen ab:

1. Wenn es bereit ist, Daten in Amazon S3 zu speichern, sendet der Cluster eine GenerateDataKey Anforderung an AWS KMS, in der die Schlüssel-ID des KMS-Schlüssels angegeben wird, den Sie bei der Konfiguration des Clusters für die Verwendung von CSE-KMS ausgewählt haben. Die Anforderung enthält Verschlüsselungskontext; weitere Informationen finden Sie unter Verschlüsselungskontext.
2. AWS KMS generiert einen eindeutigen Datenverschlüsselungsschlüssel (Datenschlüssel) und sendet dann zwei Kopien dieses Datenschlüssels an den Cluster. Eine Kopie ist unverschlüsselt (Klartext) und die andere Kopie ist mit dem KMS-Schlüssel verschlüsselt.
3. Der Cluster verwendet den Klartext-Datenschlüssel, um die Daten zu verschlüsseln, und entfernt den Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.
4. Der Cluster kombiniert die verschlüsselten Daten und eine verschlüsselte Kopie des Datenschlüssels in einem einzelnen verschlüsselten Objekt.
5. Der Cluster sendet das verschlüsselte Objekt zum Speichern an Amazon S3.

Die Entschlüsselung funktioniert wie folgt:

1. Der Cluster fordert das verschlüsselte Datenobjekt von einem S3-Bucket an.
2. Amazon S3 sendet das verschlüsselte Objekt an den Cluster.
3. Der Cluster extrahiert den verschlüsselte Datenschlüssel aus dem verschlüsselten Objekt und sendet dann den verschlüsselten Datenschlüssel mit einer AWS KMSEntschlüsselungsanforderung an . Die Anforderung enthält einen Verschlüsselungskontext.
4. AWS KMS entschlüsselt den verschlüsselten Datenschlüssel mit dem gleichen KMS-Schlüssel, der für die Verschlüsselung verwendet wurde, und sendet dann den entschlüsselten (Klartext) Datenschlüssel an den Cluster.
5. Der Cluster verwendet den Klartext-Datenschlüssel, um die verschlüsselten Daten zu entschlüsseln, und entfernt den Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.

Verschlüsseln von Daten auf den Speicher-Volumes von Cluster-Knoten

Ein Amazon-EMR-Cluster ist eine Sammlung von Amazon Elastic Compute Cloud (Amazon EC2)-Instances. Jede Instance in einem Cluster wird als Cluster-Knoten oder Knoten bezeichnet. Jeder Knoten kann zwei Arten von Speicher-Volumes besitzen: Instance-Speicher-Volumes und Amazon Elastic Block Store (Amazon EBS)-Volumes. Sie können den Cluster auf die Verwendung von Linux Unified Key Setup (LUKS) einstellen, um beide Arten von Speicher-Volumen auf dem Knoten zu verschlüsseln (aber nicht das Start-Volume jedes Knotens). Dies wird als lokale Laufwerksverschlüsselung bezeichnet.

Wenn Sie die lokale Laufwerksverschlüsselung für einen Cluster aktivieren, können Sie den LUKS-Schlüssel mit einem KMS-Schlüssel in AWS KMS verschlüsseln. Sie müssen einen kundenverwalteten Schlüssel auswählen, den Sie erstellen. Sie können keinen Von AWS verwalteter Schlüssel verwenden. Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie sicherstellen, dass der Amazon-EMR-Cluster über die Berechtigung zur Nutzung des KMS-Schlüssels verfügt. Weitere Information erhalten Sie unter Verschlüsselung von Daten mit AWS KMS keys im Amazon-EMR-Management-Handbuch.

Wenn Sie die lokale Laufwerksverschlüsselung mit einem KMS-Schlüssel aktivieren, läuft der Verschlüsselungsprozess folgendermaßen ab:

1. Wenn jeder Cluster-Knoten gestartet wird, sendet er eine GenerateDataKey Anforderung an AWS KMS, in der die Schlüssel-ID des KMS-Schlüssels angegeben wird, den Sie ausgewählt haben, als Sie die lokale Festplattenverschlüsselung für den Cluster aktiviert haben.
2. AWS KMS generiert einen eindeutigen Datenverschlüsselungsschlüssel (Datenschlüssel) und sendet dann zwei Kopien dieses Datenschlüssels an den Knoten. Eine Kopie ist unverschlüsselt (Klartext) und die andere Kopie ist unter dem KMS-Schlüssel verschlüsselt.
3. Der Knoten verwendet eine base64-verschlüsselte Version des Klartext-Datenschlüssels als Passwort, um den LUKS-Schlüssel zu schützen. Der Knoten speichert die verschlüsselte Kopie des Datenschlüssel in seinem Start-Volume.
4. Wenn der Knoten neu gestartet wird, sendet der neu gestartete Knoten den verschlüsselten Datenschlüssel mit einer AWS KMSDecrypt-Anforderung an .
5. AWS KMS entschlüsselt den verschlüsselten Datenschlüssel mit dem gleichen KMS-Schlüssel, der für die Verschlüsselung verwendet wurde, und sendet den entschlüsselten (Klartext) Datenschlüssel an den Knoten.
6. Der Knoten verwendet die base64-verschlüsselte Version des Klartext-Datenschlüssels als Passwort, um den LUKS-Schlüssel freizuschalten.

Verschlüsselungskontext

Jeder in AWS KMS integrierte AWS-Service kann einen Verschlüsselungskontext angeben, wenn er AWS KMS verwendet, um Datenschlüssel zu erstellen bzw. um Daten zu verschlüsseln oder zu entschlüsseln. Der Verschlüsselungskontext enthält zusätzliche authentifizierte Informationen, anhand derer AWS KMS die Datenintegrität überprüft. Wenn ein Service für Verschlüsselungsoperation ein Verschlüsselungskontext angibt, muss der Service denselben Verschlüsselungskontext auch für die entsprechende Entschlüsselungsoperation angeben. Andernfalls schlägt die Entschlüsselung fehl. Der Verschlüsselungskontext wird zudem in AWS CloudTrail-Protokolldateien geschrieben, sodass Sie jederzeit nachvollziehen können, warum ein bestimmter KMS-Schlüssel verwendet wurde.

Der folgende Abschnitt erläutert den Verschlüsselungskontext, der in jedem Amazon-EMR-Verschlüsselungsszenario verwendet wird, das einen KMS-Schlüssel nutzt.

Verschlüsselungskontext für die EMRFS-Verschlüsselung mit SSE-KMS

Bei SSE-KMS sendet der Amazon-EMR-Cluster Daten an Amazon S3 und Amazon S3 verwendet dann einen KMS-Schlüssel, um die Daten vor dem Speichern in einem S3-Bucket zu verschlüsseln. In diesem Fall verwendet Amazon S3 den Amazon-Ressourcennamen (ARN) des S3-Objekts als Verschlüsselungskontext für jede - GenerateDataKey und Decrypt-Anforderung, die an gesendet wirdAWS KMS. Das folgende Beispiel zeigt eine JSON-Darstellung des von Amazon S3 verwendeten Verschlüsselungskontextes:

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Verschlüsselungskontext für die EMRFS-Verschlüsselung mit CSE-KMS

Bei CSE-KMS verwendet der Amazon-EMR-Cluster einen KMS-Schlüssel, um die Daten vor dem Senden an Amazon S3 für die Speicherung zu verschlüsseln. In diesem Fall verwendet der Cluster den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels als Verschlüsselungskontext für jede - GenerateDataKey und Decrypt-Anforderung, die er an sendetAWS KMS. Das folgende Beispiel zeigt eine JSON-Darstellung des vom Cluster verwendeten Verschlüsselungskontextes.

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Verschlüsselungskontext für die Laufwerksverschlüsselung mit LUKS

Wenn ein Amazon-EMR-Cluster die lokale Festplattenverschlüsselung mit LUKS verwendet, geben die Cluster-Knoten keinen Verschlüsselungskontext mit den Anforderungen GenerateDataKey und Decrypt an, die sie an sendenAWS KMS.

WieAWS Nitro Enclaves AWS KMS nutzt

AWS KMS unterstützt kryptografische Bescheinigungen für AWS Nitro Enclaves. Anwendungen, die AWS Nitro Enclaves unterstützen, rufen die folgenden AWS KMS kryptografischen Operationen mit einem signierten Bescheinigungsdokument für die Enklave auf. Diese AWS KMS-APIs verifizieren, dass das Bescheinigungsdokument aus einer Nitro-Enklave stammt. Anstatt Klartextdaten in der Antwort zurückzugeben, verschlüsseln diese APIs den Klartext mit dem öffentlichen Schlüssel aus dem Bescheinigungsdokument und geben einen verschlüsselten Text zurück, der nur mit dem entsprechenden privaten Schlüssel in der Enklave entschlüsselt werden kann.

- Decrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

Die folgende Tabelle zeigt, wie sich die Antwort auf Nitro-Enclave-Anfragen von der Standardantwort für jeden API-Vorgang unterscheidet.

AWS KMS-Operation	Standardantwort	Antwort für AWS Nitro Enclaves
`Decrypt`	Gibt Klartextdaten zurück	Gibt die Klartextdaten zurück, die durch den öffentlichen Schlüssel aus dem Bescheinigungsdokument verschlüsselt wurden
`GenerateDataKey`	Gibt eine Klartextkopie des Datenschlüssels zurück	Liefert eine Kopie des mit dem öffentlichen Schlüssel des Bescheinigungsdokuments

AWS KMS-Operation	Standardantwort	Antwort für AWS Nitro Enclaves
	(Gibt auch eine Kopie des Datenschlüssels zurück, der durch einen KMS-Schlüssel verschlüsselt wurde)	verschlüsselten Datenschlüssels (Gibt auch eine Kopie des Datenschlüssels zurück, der durch einen KMS-Schlüssel verschlüsselt wurde)
`GenerateDataKeyPair`	Gibt eine Klartextkopie des privaten Schlüssels zurück (Gibt auch den öffentlichen Schlüssel und eine Kopie des privaten Schlüssels zurück, der mit einem KMS-Schlüssel verschlüsselt ist)	Gibt eine Kopie des privaten Schlüssels zurück, der mit dem öffentlichen Schlüssel des Bescheinigungsdokuments verschlüsselt wurde (Gibt auch den öffentlichen Schlüssel und eine Kopie des privaten Schlüssels zurück, der mit einem KMS-Schlüssel verschlüsselt ist)
`GenerateRandom`	Gibt eine zufällige Byte-Zeichenfolge zurück	Gibt die mit dem öffentlichen Schlüssel verschlüsselte zufällige Bytefolge aus dem Bescheinigungsdokument zurück

AWS KMS unterstützt Richtlinien-Bedingungsschlüssel, die Sie verwenden können, um Enklaven-Operationen mit einem AWS KMS-Schlüssel auf der Grundlage des Inhalts des Bescheinigungsdokuments zuzulassen oder zu verweigern. Sie können auch Anfragen an AWS KMS für Ihre Nitro-Enklave in Ihren AWS CloudTrail-Protokollen überwachen.

Themen

- So rufen Sie AWS KMS-APIs für eine Nitro-Enklave auf
- AWS KMS-Bedingungsschlüssel für AWS Nitro Enclaves

- Überwachung von Anfragen für Nitro-Enklaven

So rufen Sie AWS KMS-APIs für eine Nitro-Enklave auf

Um AWS KMS-APIs für eine Nitro-Enklave aufzurufen, verwenden Sie den `Recipient`-Parameter in der Anfrage, um das signierte Bestätigungsdokument für die Enklave und den Verschlüsselungsalgorithmus bereitzustellen, der mit dem öffentlichen Schlüssel der Enklave verwendet werden soll. Wenn eine Anfrage den `Recipient`-Parameter mit einem signierten Beglaubigungsdokument enthält, enthält die Antwort ein `CiphertextForRecipient`-Feld mit dem Geheimtext, der mit dem öffentlichen Schlüssel verschlüsselt ist. Das Klartextfeld ist Null oder leer.

Der `Recipient`-Parameter muss ein signiertes Beglaubigungsdokument aus einer AWS-Nitro-Enklave angeben. AWS KMS stützt sich auf die digitale Signatur für das Bescheinigungsdokument der Enklave, um nachzuweisen, dass der öffentliche Schlüssel in der Anforderung aus einer gültigen Enklave stammt. Sie können kein eigenes Zertifikat angeben, um das Bescheinigungsdokument digital zu signieren.

Verwenden Sie zur Angabe des `Recipient`-Parameters das AWS Nitro Enclaves SDK oder ein beliebiges AWS-SDK. Das AWS Nitro Enclaves SDK, das nur innerhalb einer Nitro-Enklave unterstützt wird, fügt den `Recipient`-Parameter und seine Werte automatisch zu jeder AWS KMS-Anfrage hinzu. Um Anfragen für Nitro-Enklaven in den AWS-SDKs zu stellen, müssen Sie den `Recipient`-Parameter und seine Werte angeben. Die Unterstützung für die kryptografische Nitro-Enclave-Bestätigung in den AWS-SDKs wurde im März 2023 eingeführt.

AWS KMS unterstützt Richtlinien-Bedingungsschlüssel, die Sie verwenden können, um Enklaven-Operationen mit einem AWS KMS-Schlüssel auf der Grundlage des Inhalts des Bescheinigungsdokuments zuzulassen oder zu verweigern. Sie können auch Anfragen an AWS KMS für Ihre Nitro-Enklave in Ihren AWS CloudTrail-Protokollen überwachen.

Ausführliche Informationen zum `Recipient` Parameter und zum AWS-`CiphertextForRecipient`Antwortfeld finden Sie in den GenerateRandom Themen Decrypt , GenerateDataKeyGenerateDataKeyPair, und in der APIAWS Key Management Service-Referenz , dem AWS Nitro Enclaves SDK oder einem beliebigen AWS SDK. Informationen zum Einrichten der Daten und Datenschlüssel für die Verschlüsselung finden Sie unter Verwenden der kryptografischen Bescheinigung mit AWS KMS.

AWS KMS-Bedingungsschlüssel für AWS Nitro Enclaves

Sie können Bedingungsschlüssel in den Schlüsselrichtlinien und IAM-Richtlinien angeben, die den Zugriff auf Ihre AWS KMS-Ressourcen steuern. Richtlinienerklärungen, die einen Bedingungsschlüssel enthalten, sind nur wirksam, wenn die Bedingungen erfüllt sind.

AWS KMS stellt Bedingungsschlüssel bereit, die die Berechtigungen für die GenerateRandom Operationen Decrypt , GenerateDataKeyGenerateDataKeyPair, und basierend auf dem Inhalt des signierten Bescheinigungsdokuments in der Anforderung einschränken. Diese Bedingungsschlüssel funktionieren nur, wenn eine AWS KMS-Anforderung für einen Vorgang den `Recipient`-Parameter mit einem gültigen Bestätigungsdokument aus einer AWS-Nitro-Enklave enthält. Verwenden Sie zur Angabe des `Recipient`-Parameters das AWS Nitro Enclaves SDK oder ein beliebiges AWS-SDK.

Die enklavenspezifischen AWS KMS-Bedingungsschlüssel sind in Schlüsselrichtlinien-Anweisungen und IAM-Richtlinienanweisungen gültig, auch wenn sie nicht in der IAM-Konsole oder in der IAM-Dienstberechtigungsreferenz erscheinen.

kms:RecipientAttestation:ImageSha384

AWS KMS-Bedingungsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:RecipientAttestation:ImageSha384`	String	Einzelwertig	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:RecipientAttestation:ImageSha384` steuert den Zugriff auf `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair`, und `GenerateRandom` mit einem KMS-Schlüssel nur dann, wenn der Image Digest aus dem signierten Bescheinigungsdokument in der Anfrage mit dem Wert im Bedingungsschlüssel übereinstimmt. Der `ImageSha384`-Wert entspricht PCR0 im Bescheinigungsdokument. Dieser Bedingungsschlüssel ist nur wirksam, wenn

der `Recipient`-Parameter in der Anforderung ein signiertes Bestätigungsdokument für eine AWS-Nitro-Enklave angibt.

Dieser Wert ist auch in CloudTrail Ereignissen für Anfragen an AWS KMS für Nitro-Enklaven enthalten.

> **Note**
>
> Dieser Bedingungsschlüssel ist in Schlüsselrichtlinien-Anweisungen und IAM-Richtlinienanweisungen gültig, obwohl er nicht in der IAM-Konsole oder in der IAM-Serviceautorisierungsreferenz vorkommt.

Die folgende Schlüsselrichtlinienanweisung erlaubt es der `data-processing` Rolle beispielsweise, den KMS-Schlüssel für die Operationen , GenerateDataKeyGenerateDataKeyPair, und zu verwenden. GenerateRandom Der Bedingungsschlüssel `kms:RecipientAttestation:ImageSha384` erlaubt die Operationen nur, wenn der Bild-Digest-Wert (PCR0) des Bescheinigungsdokuments in der Anforderung mit dem Bild-Digest-Wert in der Bedingung übereinstimmt. Dieser Bedingungsschlüssel ist nur wirksam, wenn der `Recipient`-Parameter in der Anforderung ein signiertes Bestätigungsdokument für eine AWS-Nitro-Enklave angibt.

Wenn die Anforderung kein gültiges Bescheinigungsdokument aus einer AWS-Nitro-Enklave enthält, wird die Genehmigung verweigert, da diese Bedingung nicht erfüllt ist.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
```

```
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms:RecipientAttestation:PCR<PCR_ID>

AWS KMS-Bedingungsschlüssel	Bedingungstyp	Werttyp	API-Operationen	Richtlinientyp

`kms:RecipientAttestation:PCR<PCR_ID>`	String	Einzelwertig	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Schlüsselrichtlinien und IAM-Richtlinien

Der Bedingungsschlüssel `kms:RecipientAttestation:PCR<PCR_ID>` steuert den Zugriff auf `Decrypt`, `GenerateDataKey`, `GenerateDataKeyPair`, und `GenerateRandom` mit einem KMS-Schlüssel nur dann, wenn die Platform Configuration Registers (PCRs) aus dem signierten Bescheinigungsdokument in der Anforderung mit den PCRs im Bedingungsschlüssel übereinstimmen. Dieser Bedingungsschlüssel ist nur wirksam, wenn der `Recipient`-Parameter in der Anforderung ein signiertes Bestätigungsdokument aus einer AWS-Nitro-Enklave angibt.

Dieser Wert ist auch in CloudTrail Ereignissen enthalten, die Anfragen an AWS KMS für Nitro-Enklaven darstellen.

> **Note**
>
> Dieser Bedingungsschlüssel ist in Schlüsselrichtlinien-Anweisungen und IAM-Richtlinienanweisungen gültig, obwohl er nicht in der IAM-Konsole oder in der IAM-Serviceautorisierungsreferenz vorkommt.

Verwenden Sie das folgende Format, um einen PCR-Wert anzugeben. Verketten Sie die PCR-ID mit dem Bedingungsschlüssel-Namen. Der PCR-Wert muss eine Hexadezimalzeichenfolge in Kleinbuchstaben von bis zu 96 Bytes sein.

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

Der folgende Bedingungsschlüssel gibt beispielsweise einen bestimmten Wert für PCR1 an, der dem Hash des Kernels entspricht, der für die Enklave und den Bootstrap-Prozess verwendet wird.

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

Die folgende Beispiel-Schlüsselrichtlinienanweisung erlaubt es der`data-processing` -Rolle, den KMS-Schlüssel für die Operation Decrypt zu verwenden.

Der Bedingungsschlüssel `kms:RecipientAttestation:PCR` in dieser Anweisung erlaubt die Produktion nur, wenn der PCR1-Wert im signierten Bescheinigungsdokument in der Anforderung mit dem `kms:RecipientAttestation:PCR1`-Wert in der Bedingung übereinstimmt. Verwenden des `StringEqualsIgnoreCase`-Richtlinienoperators, um einen Vergleich der PCR-Werte ohne Berücksichtigung der Groß-/Kleinschreibung zu erfordern.

Wenn die Anforderung kein Bescheinigungsdokument enthält, wird die Berechtigung verweigert, da diese Bedingung nicht erfüllt ist.

```
{
  "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
    }
  }
}
```

Überwachung von Anfragen für Nitro-Enklaven

Sie können Ihre -AWS CloudTrailProtokolle verwenden, um die GenerateRandom Operationen Decrypt , GenerateDataKeyGenerateDataKeyPair, und für eine -AWSNitro-Enklave zu überwachen. In diesen Protokolleinträgen enthält das `additionalEventData`-Feld ein `recipient`-Feld mit der Modul-ID (`attestationDocumentModuleId`), dem Image-Digest (`attestationDocumentEnclaveImageDigest`) und den Plattformkonfigurationsregistern (PCRs) aus dem Bestätigungsdokument in der Anfrage. Diese Felder sind nur enthalten, wenn der `Recipient`-Parameter in der Anforderung ein signiertes Bestätigungsdokument aus einer AWS-Nitro-Enklave angibt.

Die Modul-ID ist die Enklave-ID der Nitro-Enklave. Der Image-Digest ist der SHA384-Hash des Enklave-Images. Sie können den Image-Digest und die PCR-Werte als Bedingungen für Schlüsselrichtlinien und IAM-Richtlinien verwenden. Informationen zu den PCRs finden Sie im AWS-Nitro-Enclaves-Benutzerhandbuch unter Wo finde ich die Maße einer Enklave.

Dieser Abschnitt zeigt einen CloudTrail Beispielprotokolleintrag für jede der unterstützten Nitro-Enklavenanforderungen an AWS KMS.

Entschlüsseln (für eine Enklave)

Das folgende Beispiel zeigt einen AWS CloudTrail-Protokolleintrag für die Decrypt-Operation für eine AWS-Nitro-Enklave.


```
{
    "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 (für eine Enklave)

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag einer -GenerateDataKeyOperation für eine -AWSNitro-Enklave.

```
{
    "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 (für eine Enklave)

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag einer -GenerateDataKeyPairOperation für eine -AWSNitro-Enklave.

```
{
```

```
    "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 (für eine Enklave)

Das folgende Beispiel zeigt einen -AWS CloudTrailProtokolleintrag einer -GenerateRandomOperation für eine -AWSNitro-Enklave.

```
{
    "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"
}
```

Wie Amazon Redshift AWS KMS nutzt

In diesem Thema erfahren Sie, wie Amazon Redshift mithilfe von AWS KMS Daten verschlüsselt.

Themen

- Amazon-Redshift-Verschlüsselung
- Verschlüsselungskontext

Amazon-Redshift-Verschlüsselung

Ein Amazon Redshift Data Warehouse ist eine Sammlung von Computing-Ressourcen, den sogenannten Knoten, die zu Gruppen, den sogenannten Clustern, zusammengefasst werden. In jedem Cluster wird eine Amazon-Redshift-Engine ausgeführt, und er enthält mindestens eine Datenbank.

Amazon Redshift verwendet zur Verschlüsselung eine schlüsselbasierte Architektur mit vier Ebenen. Diese Architektur besteht aus Datenverschlüsselungsschlüsseln, einem Datenbankschlüssel, einem Clusterschlüssel und einem Stammschlüssel. Sie können einen AWS KMS key als Stammschlüssel verwenden.

Datenverschlüsselungsschlüssel verschlüsseln Datenblöcke im Cluster. Jedem Datenblock wird ein zufällig generierter AES-256-Schlüssel zugewiesen. Diese Schlüssel werden mithilfe des Datenbankschlüssels des Clusters verschlüsselt.

Der Datenbankschlüssel verschlüsselt Datenverschlüsselungsschlüssel im Cluster. Bei ihm handelt es sich um einen zufällig generierten AES-256-Schlüssel. Er wird auf einem Datenträger in einem separaten Netzwerk außerhalb des Amazon-Redshift-Clusters gespeichert und über einen sicheren Kanal an den Cluster übergeben.

Der Clusterschlüssel verschlüsselt den Datenbankschlüssel des Amazon Redshift-Clusters. Zur Verwaltung des Clusterschlüssels können Sie AWS KMS, AWS CloudHSM oder ein externes Hardwaresicherheitsmodul (HSM) verwenden. Weitere Details finden Sie im Dokumentationsthema Amazon Redshift-Datenbankverschlüsselung.

Anfordern können Sie die Verschlüsselung durch die Aktivierung des entsprechenden Kontrollkästchens in der Amazon-Redshift-Konsole. In der Liste unter dem Kontrollkästchen für die Verschlüsselung können Sie einen kundenverwalteten Schlüssel auswählen, der verwendet werden soll. Wenn Sie keinen kundenverwalteten Schlüssel angeben, verwendet Amazon Redshift den Von AWS verwalteter Schlüssel für Amazon Redshift unter Ihrem Konto.

> **Important**
>
> Amazon Redshift unterstützt nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können einen asymmetrischen KMS-Schlüssel nicht als in einem Amazon-Redshift-Verschlüsselungs-Workflow verwenden. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Verschlüsselungskontext

Jeder in AWS KMS integrierte Service gibt einen Verschlüsselungskontext an, wenn Daten angefordert, verschlüsselt und entschlüsselt werden. Der Verschlüsselungskontext enthält zusätzliche authentifizierte Daten (AAD), anhand derer AWS KMS die Datenintegrität überprüft. Wenn für eine Verschlüsselungsoperation ein Verschlüsselungskontext angegeben wird, gibt der Service denselben Verschlüsselungskontext auch für die Entschlüsselungsoperation an. Andernfalls schlägt die Entschlüsselung fehl. Amazon Redshift gibt die Cluster-ID und den Erstellungszeitpunkt als Verschlüsselungskontext an. Im `requestParameters` Feld einer CloudTrail Protokolldatei sieht der Verschlüsselungskontext ähnlich wie dieser aus.

```
"encryptionContext": {
    "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",
    "aws:redshift:createtime": "20150206T1832Z"
},
```

Sie können in Ihren CloudTrail Protokollen nach dem Clusternamen suchen, um zu verstehen, welche Operationen mit einem AWS KMS key (KMS-Schlüssel) ausgeführt wurden. Zu den möglichen Operationen gehören die Cluster-Verschlüsselung, die Cluster-Entschlüsselung und die Generierung von Datenschlüsseln.

Wie Amazon Relational Database Service (Amazon RDS) AWS KMS nutzt

Sie können Amazon Relational Database Service (Amazon RDS) verwenden, um eine relationale Datenbank in der Cloud einzurichten, zu betreiben und zu skalieren. Sie können Ihre Amazon-RDS-Ressourcen mit einem Von AWS verwalteter Schlüssel oder vom Kunden verwalteten Schlüssel verschlüsseln. Amazon RDS baut auf Amazon Elastic Block Store (Amazon EBS)-Verschlüsselung, um Datenbank-Volumes vollständig zu verschlüsseln.

Ausführliche Informationen darüber, wie Amazon RDS KMS-Schlüssel zum Schutz Ihrer Ressourcen verwendet, finden Sie unter Verschlüsselung von Amazon RDS-Ressourcen und AWS KMSSchlüsselverwaltung im Amazon-RDS-Benutzerhandbuch.

Wie AWS Secrets ManagerAWS KMS verwendet

AWS Secrets Manager ist ein AWS-Service zur Verschlüsselung und Speicherung von Geheimnissen. Außerdem entschlüsselt er diese transparent und gibt sie als Klartext zurück. Der Service wurde insbesondere konzipiert, um Anwendungsgeheimnisse wie Anmeldeinformationen zu speichern, die regelmäßig geändert werden und nicht hartgecodet oder als Klartext in der Anwendung gespeichert werden sollten. Statt hartkodierte Anmeldeinformationen oder Tabellen zu verwenden, ruft Ihre Anwendung Secrets Manager auf.

Secrets Manager unterstützt auch Funktionen für die regelmäßige Drehung von Geheimnissen, die häufig verwendeten Datenbanken zugeordnet sind. Außerdem werden kürzlich rotierte Geheimnisse vor dem Speichern verschlüsselt.

Secrets Manager ist in AWS Key Management Service (AWS KMS) integriert, sodass jede Version eines Geheimnisses mit einem eindeutigen Datenschlüssel verschlüsselt wird, der durch einen AWS KMS key geschützt ist. Diese Integration schützt Ihre Secrets mit Verschlüsselungsschlüsseln, die AWS KMS nie unverschlüsselt verlassen. Außerdem können Sie so benutzerdefinierte Berechtigungen für den KMS-Schlüssel festlegen und die Operationen zum Erstellen, Verschlüsseln und Entschlüsseln der Datenschlüssel prüfen, die Ihre Geheimnisse schützen.

Informationen dazu, wie Secrets Manager KMS-Schlüssel zum Schutz Ihrer Geheimnisse verwendet, finden Sie unter Verschlüsseln und Entschlüsseln von Geheimnissen im AWS Secrets Manager-Benutzerhandbuch.

Wie Amazon Simple Email Service (Amazon SES) AWS KMS nutzt.

Sie können mit Amazon Simple Email Service (Amazon SES) E-Mails empfangen und die empfangenen Nachrichten (optional) verschlüsseln, bevor Sie sie in einem von Ihnen gewählten Amazon Simple Storage Service (Amazon S3)-Bucket speichern. Wenn Sie Amazon SES zum Verschlüsseln von E-Mail-Nachrichten konfigurieren, müssen Sie den AWS KMS AWS KMS key auswählen, mit dem Amazon SES die Nachrichten verschlüsselt. Sie können den Von AWS verwalteter Schlüssel für Amazon SES (der Alias lautet `aws/ses`) oder einen symmetrischen kundenverwalteten Schlüssel auswählen, den Sie in AWS KMS erstellt haben.

> **Important**
>
> Amazon SES unterstützt nur symmetrische KMS-Schlüssel. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Ihre E-Mail-Nachrichten von Amazon SES zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Weitere Informationen zum Empfangen von E-Mails mit Amazon SES finden Sie unter Empfangen von E-Mails mit Amazon SES im Entwicklerhandbuch für Amazon Simple Email Service.

Themen

- Übersicht über die Amazon-SES-Verschlüsselung mit AWS KMS
- Amazon-SES-Verschlüsselungskontext
- Amazon SES die Berechtigung erteilen, Ihre AWS KMS key zu nutzen
- Abrufen und Entschlüsseln von E-Mail-Nachrichten

Übersicht über die Amazon-SES-Verschlüsselung mit AWS KMS

Wenn Sie Amazon SES so konfigurieren, dass empfangene E-Mail-Nachrichten vor dem Speichern in Ihrem S3-Bucket verschlüsselt werden, läuft der Prozess folgendermaßen ab:

1. Sie erstellen eine Empfangsregel für Amazon SES, in der Sie die S3-Aktion, einen S3-Bucket für die Speicherung und einen AWS KMS key für die Verschlüsselung angeben.
2. Amazon SES erhält eine E-Mail-Nachricht, die Ihrer Empfangsregel entspricht.

3. Amazon SES fordert einen eindeutige Datenschlüssel an, der mit dem KMS-Schlüssel verschlüsselt ist, den Sie in der entsprechenden Empfangsregel angegeben haben.
4. AWS KMS erstellt einen neuen Datenschlüssel, verschlüsselt diesen Datenschlüssel mit dem angegebenen KMS-Schlüssel und sendet sowohl den verschlüsselten Datenschlüssel als auch Klartextkopien des Datenschlüssels an Amazon SES.
5. Amazon SES verschlüsselt die E-Mail-Nachricht mit dem Klartext-Datenschlüssel und entfernt diesen Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.
6. Amazon SES platziert die verschlüsselte E-Mail-Nachricht und den verschlüsselten Datenschlüssel im angegebenen S3-Bucket. Dann wird der verschlüsselte Datenschlüssel im Metadatenformat zusammen mit der verschlüsselten E-Mail-Nachricht gespeichert.

Zum Ausführen von Step 3 bis Step 6 verwendet Amazon SES den von AWS bereitgestellten Amazon S3 Encryption Client. Verwenden Sie den gleichen Client, um Ihre verschlüsselten E-Mail-Nachrichten aus Amazon S3 abzurufen und zu entschlüsseln. Weitere Informationen finden Sie unter Abrufen und Entschlüsseln von E-Mail-Nachrichten.

Amazon-SES-Verschlüsselungskontext

Wenn Amazon SES einen Datenschlüssel anfordert, um Ihre empfangenen E-Mail-Nachrichten zu verschlüsseln (Step 3 in der Übersicht über die Amazon-SES-Verschlüsselung mit AWS KMS), enthält die Anforderung einen Verschlüsselungskontext. Der Verschlüsselungskontext enthält zusätzliche authentifizierte Daten (AAD), anhand derer AWS KMS die Datenintegrität sicherstellt. Der Verschlüsselungskontext wird zudem in Ihre AWS CloudTrail-Protokolldateien geschrieben, sodass Sie herausfinden können, warum ein bestimmter AWS KMS key (KMS-Schlüssel) verwendet wurde. Amazon SES verwendet den folgenden Verschlüsselungskontext:

- Die ID des AWS-Konto, in dem Sie Amazon SES für den Empfang von E-Mail-Nachrichten konfiguriert haben
- Der Regelname der Amazon-SES-Empfangsregel, mit der die S3-Aktion in der E-Mail-Nachricht aufgerufen wurde
- Die Amazon-SES-Mitteilungs-ID für die E-Mail-Nachricht

Das folgende Beispiel zeigt eine JSON-Darstellung des von Amazon SES verwendeten Verschlüsselungskontextes:

```
{
```

```
  "aws:ses:source-account": "111122223333",
  "aws:ses:rule-name": "example-receipt-rule-name",
  "aws:ses:message-id": "d6iitobk75ur44p8kdnnp7g2n800"
}
```

Amazon SES die Berechtigung erteilen, Ihre AWS KMS key zu nutzen

Um Ihre E-Mail-Nachrichten zu verschlüsseln, können Sie den Von AWS verwalteter Schlüssel in Ihrem Konto für Amazon SES (`aws/ses`) oder einen von Ihnen erstellten kundenverwalteten Schlüssel verwenden. Amazon SES verfügt bereits über die Berechtigung, den Von AWS verwalteter Schlüssel in Ihrem Namen zu nutzen. Wenn Sie jedoch einen kundenverwalteten Schlüssel angeben, müssen Sie Amazon SES die Berechtigung erteilen, Ihren KMS-Schlüssel zum Verschlüsseln Ihrer E-Mail-Nachrichten zu verwenden, wenn Sie Ihrer Amazon-SES-Empfangsregel die S3-Aktion hinzufügen.

Wenn Sie Amazon SES die Berechtigung erteilen möchten, Ihren kundenverwalteten Schlüssel zu verwenden, fügen Sie der Schlüsselrichtlinie dieses KMS-Schlüssels die folgende Anweisung hinzu:

```
{
  "Sid": "Allow SES to encrypt messages using this KMS key",
  "Effect": "Allow",
  "Principal": {"Service": "ses.amazonaws.com"},
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:ses:rule-name": false,
      "kms:EncryptionContext:aws:ses:message-id": false
    },
    "StringEquals": {"kms:EncryptionContext:aws:ses:source-account": "ACCOUNT-ID-
WITHOUT-HYPHENS"}
  }
}
```

Ersetzen Sie *ACCOUNT-ID-WITHOUT-HYPHENS* durch die 12-stellige ID des AWS-Konto, in dem Sie Amazon SES für den Empfang von E-Mail-Nachrichten konfiguriert haben. Diese Richtlinienanweisung erlaubt es Amazon SES, Daten mit diesem KMS-Schlüssel nur unter den folgenden Bedingungen zu verschlüsseln:

- Amazon SES muss `aws:ses:message-id` und `EncryptionContext` im `aws:ses:rule-name` der AWS KMS-API-Anforderungen angeben.
- Amazon SES muss `aws:ses:source-account` im `EncryptionContext` der AWS KMS-API-Anforderungen angeben und der Wert für `aws:ses:source-account` muss mit der in der Schlüsselrichtlinie angegebenen AWS-Konto-ID übereinstimmen.

Weitere Informationen über den Verschlüsselungskontext, den Amazon SES für die Verschlüsselung Ihrer E-Mail-Nachrichten verwendet, finden Sie unter Amazon-SES-Verschlüsselungskontext. Allgemeine Informationen darüber, wie AWS KMS den Verschlüsselungskontext verwendet, finden Sie unter Verschlüsselungskontext.

Abrufen und Entschlüsseln von E-Mail-Nachrichten

Amazon SES verfügt über keine Berechtigung zum Entschlüsseln Ihrer E-Mail-Nachrichten und kann diese daher nicht für Sie entschlüsseln. Sie müssen einen Code schreiben, um Ihre E-Mail-Nachrichten von Amazon S3 abzurufen und zu entschlüsseln. Um dies zu vereinfachen, verwenden Sie den Amazon S3 Encryption Client. Die folgenden AWS-SDKs umfassen den Amazon S3 Encryption Client:

- AWS SDK for Java – Siehe AmazonS3EncryptionClient und AmazonS3EncryptionClientV2 in der AWS SDK for Java-API-Referenz.
- AWS SDK for Ruby – Siehe Aws::S3::Encryption::Client in der AWS SDK for Ruby-API-Referenz.
- AWS SDK for .NET – Siehe AmazonS3EncryptionClient in der AWS SDK for .NET-API-Referenz.
- AWS SDK for Go – Siehe s3crypto in der AWS SDK for Go-API-Referenz.

Der Amazon S3 Encryption Client vereinfacht das Verfassen der erforderlichen Anforderungen an Amazon S3 zum Abrufen der verschlüsselten E-Mail-Nachricht und an AWS KMS zum Entschlüsseln des verschlüsselten Datenschlüssels der Nachricht und zum Entschlüsseln der E-Mail-Nachricht. Um beispielsweise den verschlüsselten Datenschlüssel erfolgreich zu entschlüsseln, müssen Sie denselben Verschlüsselungskontext übergeben, den Amazon SES bei der Abfrage des Datenschlüssels von AWS KMS übergeben hat (Step 3 in der Übersicht über die Amazon-SES-Verschlüsselung mit AWS KMS). Der Amazon S3 Encryption Client erledigt diesen und andere Arbeitsschritte für Sie.

Beispiel-Code, der für die clientseitige Entschlüsselung den Amazon S3 Encryption Client in AWS SDK for Java verwendet, finden Sie hier:

- Verwenden eines KMS-Schlüssels, der in AWS KMS gespeichert ist im Amazon-Simple-Storage-Service-Benutzerhandbuch.
- Amazon-S3-Verschlüsselung mit AWS Key Management Service im AWS-Entwickler-Blog.

Wie Amazon Simple Storage Service (Amazon S3) AWS KMS nutzt

Amazon Simple Storage Service (Amazon S3) ist ein Objektspeicherdienst, der Daten als Objekte in Buckets speichert. Buckets und die darin enthaltenen Objekte sind privat und können nur zugegriffen werden, wenn Sie explizit Zugriffsberechtigungen erteilen.

Amazon S3 lässt sich in AWS Key Management Service (AWS KMS) integrieren, um serverseitige Verschlüsselung von Amazon-S3-Objekten bereitzustellen. Amazon S3 verwendet AWS KMS-Schlüssel, um Ihre Amazon S3-Objekte zu verschlüsseln. Die Verschlüsselungsschlüssel, die Ihre Objekte schützen, verlassen AWS KMS niemals unverschlüsselt. Außerdem können Sie mit dieser Integration so benutzerdefinierte Berechtigungen für den AWS KMS-Schlüssel festlegen und die Operationen zum Erstellen, Verschlüsseln und Entschlüsseln der Datenschlüssel prüfen, die Ihre Geheimnisse schützen.

Um das Volumen der Amazon S3-Aufrufe an zu reduzierenAWS KMS, verwenden Sie Amazon S3-Bucket-Schlüssel, die KMS-Schlüssel geschützt sind key-encryption-keys und für einen begrenzten Zeitraum in Amazon S3 wiederverwendet werden. Bucket-Schlüssel können die Kosten für AWS KMS-Anfragen um bis zu 99 Prozent verringern. Sie können einen Bucket-Schlüssel für alle Objekte in einem Amazon-S3-Bucket oder für ein bestimmtes Objekt in einem Amazon-S3-Bucket konfigurieren.

Weitere Informationen darüber, wie Amazon S3 AWS KMS verwendet, finden Sie unter Schutz von Daten durch serverseitige Verschlüsselung mit KMS-Schlüsseln (SSE-KMS) im Amazon S3-Benutzerhandbuch.

Wie AWS Systems Manager Parameter Store AWS KMS nutzt

Mit AWS Systems Manager Parameter Store können Sie sicherer Stringparameter erstellen. Dabei handelt es sich um Parameter mit einem Klartext-Parameternamen und einem verschlüsseltem Parameterwert. Parameter Store nutzt AWS KMSzur Verschlüsselung und Entschlüsselung der Parameterwerte von sicheren Stringparametern.

Mit Parameter Store können Sie Daten als Parameter mit Werten erstellen, speichern und verwalten. Sie können einen Parameter in Parameter Store erstellen und in mehreren Anwendungen und

Services verwenden, die den von Ihnen entworfenen Richtlinien und Berechtigungen unterliegen. Wenn Sie einen Parameterwert ändern müssen, ändern Sie nur eine Instance, anstatt fehleranfällige Änderungen an verschiedenen Quellen durchzuführen. Parameter Store unterstützt eine hierarchische Struktur für Parameternamen, sodass Sie einen Parameter für spezielle Zwecke qualifizieren können.

Um vertrauliche Daten zu verwalten, können Sie sichere Stringparameter generieren. Parameter Store verwendet AWS KMS keys, um die Parameterwerte von sicheren Stringparametern zu verschlüsseln, wenn Sie diese erstellen oder ändern. Zudem werden KMS-Schlüssel zum Entschlüsseln der Parameterwerte eingesetzt, wenn Sie auf diese zugreifen. Sie können den Von AWS verwalteter Schlüssel verwenden, den Parameter Store für Ihr Konto erstellt, oder Ihren eigenen kundenverwalteten Schlüssel angeben.

> **Important**
>
> Parameter Store unterstützt nur symmetrische KMS-Schlüssel. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Ihre Parameter zu verschlüsseln. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Parameter Store unterstützt zwei Stufen von sicheren Stringparametern: Standard und Advanced. Standard-Parameter, die 4096 Bytes nicht überschreiten können, werden direkt mit dem von Ihnen angegebenen KMS-Schlüssel verschlüsselt und entschlüsselt. Zum Verschlüsseln und Entschlüsseln von sicheren Advanced-Stringparametern verwendet Parameter Store Envelope-Verschlüsselung mit dem AWS Encryption SDK. Es ist möglich, einen sicheren Standard-Stringparameter in einen Advanced-Parameter zu konvertieren, aber nicht einen Advanced Parameter in einen Standard-Parameter. Weitere Informationen über den Unterschied zwischen den sicheren Stringparametern Standard und Advanced finden Sie unter Systems Manager Advanced Parameter im AWS Systems Manager-Benutzerhandbuch.

Themen

- Schützen von sicheren Standard-String-Parametern
- Schützen von sicheren erweiterten String-Parametern
- Festlegen der Berechtigungen zum Verschlüsseln und Entschlüsseln von Parameterwerten
- Parameter-Store-Verschlüsselungskontext
- Beheben von KMS-Schlüsselproblemen in Parameter Store

Schützen von sicheren Standard-String-Parametern

Parameter Store führt keine kryptografischen Operationen durch. Stattdessen nutzt es AWS KMS zur Verschlüsselung und Entschlüsselung von sicheren Stringparameter-Werten. Wenn Sie den Wert eines standardmäßigen, sicheren Stringparameters erstellen oder ändern, ruft Parameter Store die AWS KMS-Operation Encrypt auf. Diese Operation verwendet einen KMS-Schlüssel mit symmetrischer Verschlüsselung direkt zum Verschlüsseln des Parameterwerts, statt mit dem KMS-Schlüssel einen Datenschlüssel zu generieren.

Sie können den KMS-Schlüssel auswählen, den Parameter Store zum Verschlüsseln des Parameterwerts nutzt. Wenn Sie keinen KMS-Schlüssel angeben, verwendet Parameter Store den Von AWS verwalteter Schlüssel, den Systems Manager automatisch in Ihrem Konto erstellt. Der KMS-Schlüssel hat den Alias `aws/ssm`.

Um den Standard`aws/ssm`-KMS-Schlüssel für Ihr Konto anzuzeigen, verwenden Sie die -DescribeKeyOperation in der AWS KMS-API. In dem folgenden Beispiel wird der `describe-key`-Befehl in der AWS Command Line Interface (AWS CLI) mit dem Aliasnamen `aws/ssm` verwendet.

```
aws kms describe-key --key-id alias/aws/ssm
```

Um einen sicheren Standard-Stringparameter zu erstellen, verwenden Sie die -PutParameterOperation in der Systems Manager API. Lassen Sie den Parameter `Tier` weg oder geben Sie als Wert `Standard` ein,wobei es sich um den Standardwert handelt. Schließen Sie einen `Type`-Parameter mit einem Wert von `SecureString` ein. Nutzen Sie zum Angeben eines KMS-Schlüssels den `KeyId`-Parameter. Der Standardwert ist der Von AWS verwalteter Schlüssel für Ihr Konto, `aws/ssm`.

Parameter Store ruft dann die AWS KMS-Operation `Encrypt` mit dem KMS-Schlüssel und dem Klartext-Parameter-Wert auf. AWS KMS gibt den verschlüsselten Parameter-Wert zurück, den mit dem Parameter-Namen speichert.

Im folgenden Beispiel wird der Systems-Manager-Befehl put-parameter und dessen `--type`-Parameter in der AWS CLI zum Erstellen eines soicheren Stringparameters verwendet. Da der Befehl die optionalen Parameter `--tier` und `--key-id` weglässt, erstellt Parameter Store einen sicheren Standard-Stringparameter und verschlüsselt ihn mit dem Von AWS verwalteter Schlüssel.

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString
```

Im folgenden ähnlichen Beispiel wird der `--key-id`-Parameter zur Angabe eines kundenverwalteten Schlüssels verwendet. Im Beispiel wird eine KMS-Schlüssel-ID verwendet, um den KMS-Schlüssel zu identifizieren, Sie können jedoch einen beliebigen gültigen KMS-Schlüsselbezeichner verwenden. Da der Befehl den `Tier`-Parameter (`--tier`) weglässt, erstellt Parameter Store einen sicheren Standard-Stringparameter und keinen Advanced.

```
aws ssm put-parameter --name param1 --value "secret" --type SecureString --key-id
 1234abcd-12ab-34cd-56ef-1234567890ab
```

Wenn Sie einen sicheren Stringparameter von Parameter Store erhalten, ist dessen Wert verschlüsselt. Um einen Parameter abzurufen, verwenden Sie die -GetParameter Operation in der Systems Manager API.

Im folgenden Beispiel wird der Systems-Manager-Befehl get-parameter in der AWS CLI zum Abrufen des `MyParameter`-Parameters von Parameter Store verwendet, ohne dessen Wert zu verschlüsseln.

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

Zum Entschlüsseln des Parameterwerts vor der Rückgabe setzen Sie den `WithDecryption`-Parameter `GetParameter` auf `true`. Wenn Sie verwenden, ruft Parameter Store die AWS KMS-Operation Decrypt in Ihrem Namen auf, um den Parameter-Wert zu entschlüsseln. Infolgedessen gibt die `GetParameter`-Anforderung den Parameter mit einem Klartext-Parameterwert zurück, wie im folgenden Beispiel gezeigt.

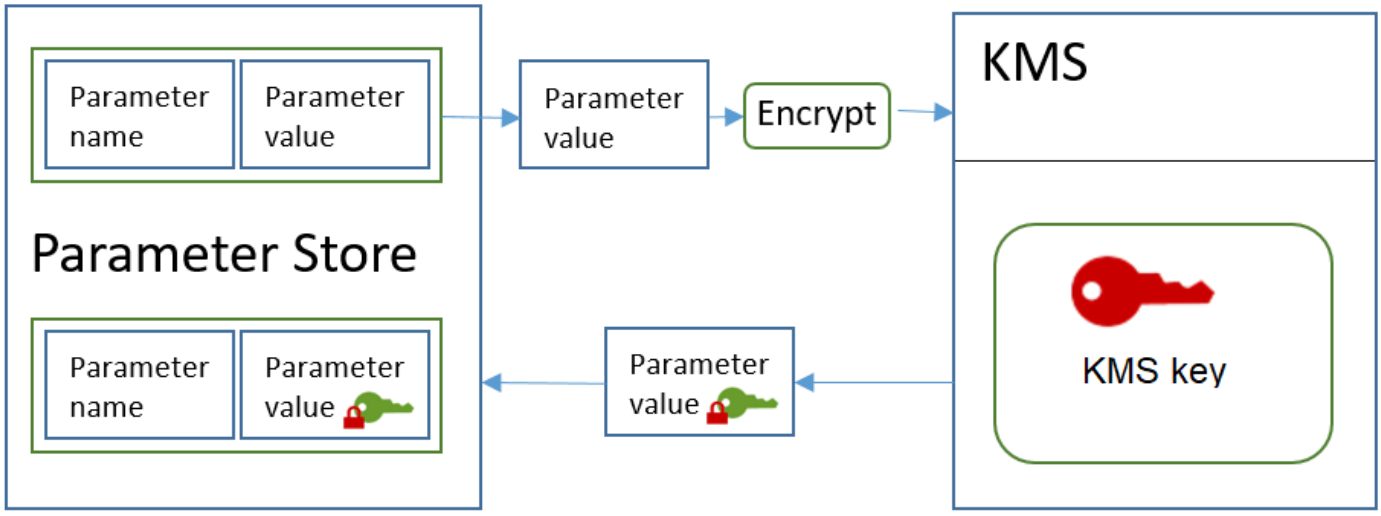
```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
```

```
}
```

Der folgende Workflow zeigt, wie Parameter Store einen KMS-Schlüssel zum Verschlüsseln und Entschlüsseln eines sicheren Standard-Stringparameters verwendet.

Verschlüsseln eines Standardparameters

1. Wenn Sie `PutParameter` verwenden, um einen sicheren Stringparameter zu erstellen, sendet Parameter Store eine `Encrypt`-Anforderung an AWS KMS. Diese Anforderung enthält den Klartext-Parameterwert, den von Ihnen ausgewählten KMS-Schlüssel und den Parameter-Store-Verschlüsselungskontext. Während der Übertragung an AWS KMS ist der Klartextwert im sicheren Stringparameter durch die Transport Layer Security (TLS) geschützt.
2. AWS KMS verschlüsselt den Parameterwert mit dem angegebenen KMS-Schlüssel und Verschlüsselungskontext. Der Chiffretext wird an Parameter Store zurückgegeben. Dort werden der Parametername und dessen verschlüsselter Wert gespeichert.



Entschlüsseln eines Standardparameters

1. Wenn Sie den `WithDecryption`-Parameter in eine `GetParameter`-Anforderung einschließen, sendet Parameter Store eine `Decrypt`-Anforderung an AWS KMS mit dem verschlüsselten Wert des sicheren Stringparameters.
2. AWS KMS verwendet denselben KMS-Schlüssel und den bereitgestellten Verschlüsselungskontext zum Entschlüsseln des verschlüsselten Werts. Es gibt den Klartext-Parameterwert (entschlüsselt) an Parameter Store zurück. Während der Übertragung sind die Klartextdaten durch TLS geschützt.

3. Parameter Store gibt den Klartext-Parameterwert an Sie in der `GetParameter`-Antwort zurück.

Schützen von sicheren erweiterten String-Parametern

Wenn Sie mit `PutParameter` einen sicheren Advanced-Stringparameter erstellen, verwendet Parameter Store zum Schutz des Parameterwerts Envelope-Verschlüsselung mit dem AWS Encryption SDK und einem AWS KMS key mit symmetrischer Verschlüsselung. Jeder erweiterte Parameterwert ist mit einem eindeutigen Datenschlüssel verschlüsselt, der wiederum mit einem KMS-Schlüssel verschlüsselt ist. Sie können den ??? für das Konto (`aws/ssm`) oder einen beliebigen kundenverwalteten Schlüssel verwenden.

Das AWS Encryption SDK ist eine clientseitige Open-Source-Bibliothek, mit der Sie Daten mithilfe von Branchenstandards und bewährten Methoden leichter verschlüsseln und entschlüsseln können. Sie wird auf mehreren Plattformen und in mehreren Programmiersprachen, einschließlich einer Befehlszeilenschnittstelle, unterstützt. Sie können den Quellcode anzeigen und zu seiner Entwicklung in beitragen GitHub.

Für jeden sicheren String-Parameterwert ruft Parameter Store die auf, AWS Encryption SDK um den Parameterwert mit einem eindeutigen Datenschlüssel zu verschlüsseln, den AWS KMS generiert (GenerateDataKey). Das AWS Encryption SDK gibt an Parameter Store eine verschlüsselte Nachricht mit dem verschlüsselten Parameterwert und einer verschlüsselten Kopie des eindeutigen Datenschlüssels zurück. Parameter Store speichert die gesamte verschlüsselte Nachricht im sicheren Stringparameter-Wert. Wenn Sie dann einen sicheren Advanced-Stringparameter-Wert abrufen, verwendet Parameter Store das AWS Encryption SDK zum Verschlüsseln des Parameterwerts. Dies erfordert einen Aufruf von AWS KMS, um den verschlüsselten Datenschlüssel zu entschlüsseln.

Um einen sicheren Advanced-Stringparameter zu erstellen, verwenden Sie die -PutParameterOperation in der Systems Manager API. Stellen Sie den Wert des Parameters `Tier` auf `Advanced` ein. Schließen Sie einen `Type`-Parameter mit einem Wert von `SecureString` ein. Nutzen Sie zum Angeben eines KMS-Schlüssels den `KeyId`-Parameter. Der Standardwert ist der Von AWS verwalteter Schlüssel für Ihr Konto, `aws/ssm`.

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString --
tier Advanced
```

Im folgenden ähnlichen Beispiel wird der `--key-id`-Parameter zur Angabe eines kundenverwalteten KMS-Schlüssels verwendet. Das Beispiel verwendet den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels, aber Sie können jeden gültigen KMS-Schlüsselbezeichner angeben.

```
aws ssm put-parameter --name MyParameter --value "secret_value"
 --type SecureString --tier Advanced --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Wenn Sie einen sicheren Stringparameter aus Parameter Store abrufen, ist sein Wert die verschlüsselte Nachricht, die von AWS Encryption SDK zurückgegeben wird. Um einen Parameter abzurufen, verwenden Sie die -GetParameter Operation in der Systems Manager API.

Das folgende Beispiel verwendet die Systems-Manager-Operation `GetParameter` zum Anfordern des `MyParameter`-Parameters von Parameter Store, ohne dessen Wert zu entschlüsseln.

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

Zum Entschlüsseln des Parameterwerts vor der Rückgabe setzen Sie den `WithDecryption`-Parameter `GetParameter` auf `true`. Wenn Sie verwenden, ruft Parameter Store die AWS KMS-Operation Decrypt in Ihrem Namen auf, um den Parameter-Wert zu entschlüsseln. Infolgedessen gibt die `GetParameter`-Anforderung den Parameter mit einem Klartext-Parameterwert zurück, wie im folgenden Beispiel gezeigt.

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Es ist nicht möglich, einen sicheren Advanced-Stringparameter in einen Standard-Parameter zu konvertieren, aber es ist möglich einen sicheren Standard-Stringparameter in einen Advanced-

Parameter zu konvertieren. Um einen sicheren Standard-Stringparameter in einen sicheren Advanced-Stringparameter zu konvertieren, verwenden Sie die `PutParameter`-Operation mit dem `Overwrite`-Parameter. `Type` muss den Wert `SecureString` und `Tier` muss den Wert `Advanced` haben. Der `KeyId`-Parameter, der einen kundenverwalteten Schlüssel identifiziert, ist optional. Wenn Sie ihn weglassen, verwendet Parameter Store den Von AWS verwalteter Schlüssel für das Konto. Sie können jeden KMS-Schlüssel angeben, den der Prinzipal verwenden darf, selbst wenn der Standardparameter von Ihnen mit einem anderen KMS-Schlüssel verschlüsselt wurde.

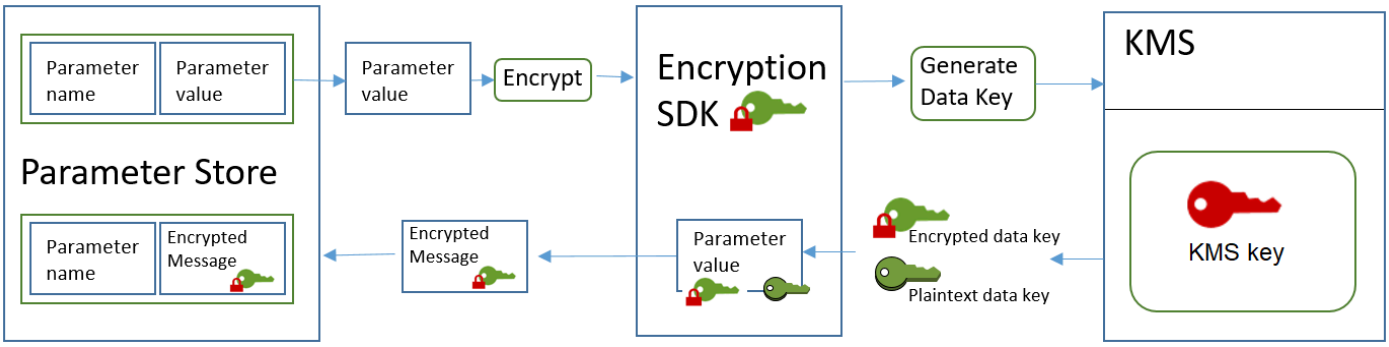
Wenn Sie den `Overwrite`-Parameter verwenden, verwendet Parameter Store das AWS Encryption SDKum den Parameterwert zu verschlüsseln. Anschließend wird die neu verschlüsselte Nachricht in Parameter Store gespeichert.

```
$ aws ssm put-parameter --name myStdParameter --value "secret_value"  --type
 SecureString --tier Advanced --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --overwrite
```

Der folgende Workflow zeigt, wie Parameter Store einen KMS-Schlüssel zum Verschlüsseln und Entschlüsseln eines sicheren Advanced-Stringparameter verwendet.

Verschlüsseln eines erweiterten Parameters

1. Wenn Sie mit `PutParameter` einen sicheren Advanced-Stringparameter erstellen, verwendet Parameter Store das AWS Encryption SDK und AWS KMS, um den Parameterwert zu verschlüsseln Parameter Store ruft die AWS Encryption SDK mit dem Parameter-Wert, dem von Ihnen angegebenen KMS-Schlüssel und dem Parameter-Store-Verschlüsselungskontext auf.
2. AWS Encryption SDK sendet eine GenerateDataKey Anforderung an AWS KMS mit der ID des von Ihnen angegebenen KMS-Schlüssels und dem Parameter Store-Verschlüsselungskontext. AWS KMS gibt zwei Kopien des eindeutigen Datenschlüssels zurück: eine Kopie im Klartext und eine mit dem KMS-Schlüssel verschlüsselte Kopie. (Der Verschlüsselungskontext wird beim Verschlüsseln des Datenschlüssels verwendet.)
3. Das AWS Encryption SDK verschlüsselt den Parameterwert mithilfe des Klartext-Datenschlüssels. Es wird eine verschlüsselte Nachricht mit dem verschlüsselten Parameterwert, dem verschlüsselten Datenschlüssel und anderen Daten, einschließlich des Parameter-Store-Verschlüsselungskontextes, zurückgegeben.
4. Parameter Store speichert die verschlüsselte Nachricht als Parameterwert.



Entschlüsseln eines erweiterten Parameters

1. Sie können den `WithDecryption`-Parameter in eine `GetParameter`-Anforderung einschließen, um einen sicheren Advanced-Stringparameter zu erhalten. In diesem Fall übergibt Parameter Store die verschlüsselte Nachricht aus dem Parameterwert an eine Entschlüsselungsmethode des AWS Encryption SDK.
2. Das AWS Encryption SDK ruft die AWS KMS-Operation Decrypt auf. Es übergibt den verschlüsselten Datenschlüssel und den Parameter-Store-Verschlüsselungskontext aus der verschlüsselten Nachricht.
3. AWS KMS verwendet den KMS-Schlüssel und den Parameter-Store-Verschlüsselungskontext, um den verschlüsselten Datenschlüssel zu entschlüsseln. Anschließend gibt die Operation den Klartext-Datenschlüssel (entschlüsselt) an das AWS Encryption SDK zurück.
4. Das AWS Encryption SDK verwendet den Klartext-Datenschlüssel zum Entschlüsseln des Parameterwertes. Es gibt den Klartext-Parameterwert an Parameter Store zurück.
5. Parameter Store überprüft den Verschlüsselungskontext und gibt den Klartext-Parameterwert in der `GetParameter`-Antwort an Sie zurück.

Festlegen der Berechtigungen zum Verschlüsseln und Entschlüsseln von Parameterwerten

Zum Verschlüsseln eines sicheren Standard-Stringparameter-Werts benötigt der Benutzer die `kms:Encrypt`-Berechtigung. Zum Verschlüsseln eines sicheren Advanced-Stringparameter-Werts benötigt der Benutzer die `kms:GenerateDataKey`-Berechtigung. Zum Entschlüsseln des sicheren Stringparameter-Werts beider Typen benötigt der Benutzer die `kms:Decrypt`-Berechtigung.

Sie können mit IAM-Richtlinien die Berechtigungen für Benutzer zum Aufrufen der Systems-Manager-Operationen `PutParameter` und `GetParameter` gewähren oder verweigern.

Wenn Sie Ihre sicheren Stringparameter-Werte mit benutzerverwalteten Schlüsseln verschlüsseln, können Sie zum Verwalten der Verschlüsselungs- und Entschlüsselungs-Berechtigungen IAM-Richtlinien und Schlüsselrichtlinien verwenden. Sie können jedoch keine Zugriffssteuerungs-Richtlinien für den standardmäßigen `aws/ssm`-KMS-Schlüssel einrichten. Ausführliche Informationen zur Steuerung des Zugriffs auf kundenverwaltete Schlüssel finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS.

Das folgende Beispiel zeigt eine IAM-Richtlinie, die für sichere Standard-Stringparameter bestimmt ist. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `PutParameter` für alle Parameter im Pfad `FinancialParameters`. Die Richtlinie ermöglicht dem Benutzer auch, die AWS KMS-Operation `Encrypt` für einen kundenverwalteten Beispiel-KMS-Schlüssel aufzurufen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/
FinancialParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Das folgende Beispiel zeigt eine IAM-Richtlinie an, die für sichere Advanced-Stringparameter bestimmt ist. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `PutParameter` für alle Parameter im Pfad `ReservedParameters`. Die Richtlinie ermöglicht dem Benutzer auch, die AWS KMS-Operation `GenerateDataKey` für einen kundenverwalteten Beispiel-KMS-Schlüssel aufzurufen.

```
{
```

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

Das letzte Beispiel zeigt eine IAM-Richtlinie, die für sichere Stringparameter beider Typen verwendet werden kann. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `GetParameter` (und verwandte Operationen) für alle Parameter im Pfad `ITParameters`. Die Richtlinie ermöglicht dem Benutzer auch, die AWS KMS-Operation `Decrypt` für einen kundenverwalteten Beispiel-KMS-Schlüssel aufzurufen.

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

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

Parameter-Store-Verschlüsselungskontext

Ein Verschlüsselungskontext ist eine Gruppe von Schlüssel/Wert-Paaren mit willkürlichen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext kryptografisch an die verschlüsselten Daten. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Sie können den Verschlüsselungskontext auch nutzen, um eine kryptografische Operation in Audit-Datensätzen und -Protokollen zu identifizieren. Die Verschlüsselungskontext wird in Klartext-Protokollen wie AWS CloudTrail angezeigt.

Das AWS Encryption SDK nimmt auch einen Verschlüsselungskontext, dieser wird allerdings anders verarbeitet. Parameter Store stellt den Verschlüsselungskontext für die Verschlüsselungsmethode bereit. Das AWS Encryption SDK bindet den Verschlüsselungskontext kryptographisch an die verschlüsselten Daten. Es schließt den Verschlüsselungskontext als Klartext im Header der von ihm zurückgegebenen verschlüsselten Nachricht ein. Anders als bei AWS KMS nehmen die AWS Encryption SDK-Entschlüsselungsmethoden keinen Verschlüsselungskontext als Eingabe an. Stattdessen erhält das AWS Encryption SDK beim Entschlüsseln von Daten den Verschlüsselungskontext aus der verschlüsselten Nachricht. Parameter Store überprüft den Verschlüsselungskontext und fügt den erwarteten Wert hinzu, bevor es den Klartext-Parameterwert an Sie zurückgibt.

Parameter Store verwendet den folgenden Verschlüsselungskontext in seinen kryptografischen Operationen:

- Schlüssel: `PARAMETER_ARN`
- Wert: Der Amazon-Ressourcenname (ARN) des Parameters, der verschlüsselt wird.

Das Format des Verschlüsselungskontexts sieht wie folgt aus:

```
"PARAMETER_ARN":"arn:aws:ssm:<REGION_NAME>:<ACCOUNT_ID>:parameter/<parameter-name>"
```

Parameter Store enthält beispielsweise den Verschlüsselungskontext in Aufrufen zum Verschlüsseln und Entschlüsseln des `MyParameter`-Parameters in einem Beispiel-AWS-Konto und einer Region.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
```

Befindet sich der Parameter in einem hierarchischen Parameter-Store-Pfad, sind sowohl der Pfad als auch der Name im Verschlüsselungskontext enthalten. Dieser Verschlüsselungskontext wird beispielsweise beim Verschlüsseln und Entschlüsseln des `MyParameter`-Parameters im `/ReadableParameters`-Pfad in einem Beispiel-AWS-Konto und einer Region verwendet.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

Sie können einen verschlüsselten sicheren Stringparameter entschlüsseln, indem Sie die AWS KMS-Operation `Decrypt` mit dem richtigen Verschlüsselungskontext und dem verschlüsselten Parameterwert, den die Systems-Manager-Operation `GetParameter` zurückgibt, aufrufen. Wir empfehlen jedoch, die Parameterwerte von Parameter Store mithilfe der `GetParameter`-Operation mit dem `WithDecryption`-Parameter zu entschlüsseln.

Sie können den Verschlüsselungskontext auch in eine IAM-Richtlinie einschließen. So können Sie einen Benutzer beispielsweise nur dazu berechtigen, einen bestimmten Parameterwert oder eine bestimmte Gruppe von Parameterwerten zu entschlüsseln.

Das folgende Beispiel einer IAM-Richtlinie erlaubt es dem Benutzer, den Wert des `MyParameter`-Parameters abzurufen und den Wert mit dem angegebenen KMS-Schlüssel zu entschlüsseln. Diese Berechtigungen gelten jedoch nur, wenn der Verschlüsselungskontext mit der angegebenen Zeichenfolge übereinstimmt. Diese Berechtigungen gelten nicht für andere Parameter oder KMS-Schlüssel und der Aufruf an `GetParameter` schlägt fehl, wenn der Verschlüsselungskontext nicht mit der Zeichenfolge übereinstimmt.

Bevor Sie eine Richtlinienanweisung wie diese verwenden, ersetzen Sie die Beispiels-ARNs durch gültige Werte.

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

```
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:PARAMETER_ARN":"arn:aws:ssm:us-
west-2:111122223333:parameter/MyParameter"
                }
            }
        }
    ]
}
```

Beheben von KMS-Schlüsselproblemen in Parameter Store

Um eine Operation auf einem sicheren Stringparameter durchzuführen, muss Parameter Store in der Lage sein, den AWS KMS-Schlüssel zu nutzen, den Sie für die beabsichtigte Operation angeben. Die meisten Parameter-Store-Fehler im Zusammenhang mit KMS-Schlüsseln treten aus folgenden Gründen auf:

- Die Anmeldeinformationen, die eine Anwendung verwendet, sind nicht berechtigt, eine bestimmte Aktion mit einem KMS-Schlüssel durchzuführen.

  Zur Behebung dieses Problems führen Sie die Anwendung mit anderen Anmeldeinformationen aus oder ändern die IAM- oder Schlüssel-Richtlinie, die diese Operation verhindert. Weitere Informationen zu IAM- und Schlüssel-Richtlinien in AWS KMS finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS.
- Der KMS-Schlüssel wurde nicht gefunden.

  Dies geschieht in der Regel, wenn Sie einen falschen Bezeichner für den KMS-Schlüssel verwenden. Finden Sie die richtigen Bezeichner für den KMS-Schlüssel und wiederholen Sie den Befehl.

- Der KMS-Schlüssel ist nicht aktiviert. In diesem Fall gibt Parameter Store eine InvalidKeyId Ausnahme mit einer detaillierten Fehlermeldung von zurückAWS KMS. Wenn der Schlüsselstatus des KMS-Schlüssels `Disabled` lautet, aktivieren Sie ihn. Wenn der Status `Pending Import` lautet, führen Sie das Importverfahren durch. Lautet der Schlüsselstatus `Pending Deletion`, brechen Sie die Schlüssellöschung ab oder verwenden Sie einen anderen KMS-Schlüssel.

  Wie Sie den Schlüsselstatus eines KMS-Schlüssels in der AWS KMS-Konsole und auf den Seiten zu Customer managed keys (Kundenverwaltete Schlüssel) oder Von AWS verwaltete Schlüssel finden können, erfahren Sie in der Status-Spalte. Um den Status eines KMS-Schlüssels mit der AWS KMS-API zu ermitteln, verwenden Sie die -DescribeKeyOperation.

So WorkMail verwendet Amazon AWS KMS

In diesem Thema wird erläutert, wie Amazon WorkMail verwendetAWS KMS, um E-Mail-Nachrichten zu verschlüsseln.

Themen

- Amazon- WorkMail Übersicht
- Amazon- WorkMail Verschlüsselung
- Autorisieren der Nutzung des KMS-Schlüssels
- Amazon WorkMail -Verschlüsselungskontext
- Überwachen der Amazon- WorkMail Interaktion mit AWS KMS

Amazon- WorkMail Übersicht

Amazon WorkMail ist ein sicherer, verwalteter Business-E-Mail- und Kalenderservice mit Unterstützung für bestehende Desktop- und mobile E-Mail-Clients. Sie können eine Amazon-WorkMail Organisation erstellen und ihr eine oder mehrere E-Mail-Domänen zuweisen, die Sie besitzen. Anschließend können Sie Postfächer für die E-Mail-Benutzer und Verteilergruppen in der Organisation erstellen.

Amazon verschlüsselt alle Nachrichten in den Postfächern aller Amazon- WorkMail Organisationen WorkMail transparent, bevor die Nachrichten auf die Festplatte geschrieben werden, und entschlüsselt die Nachrichten transparent, wenn Benutzer darauf zugreifen. Es gibt keine Option zum Deaktivieren der Verschlüsselung. Zum Schutz der Verschlüsselungsschlüssel, die die Nachrichten schützen, WorkMail ist Amazon in AWS Key Management Service (AWS KMS) integriert.

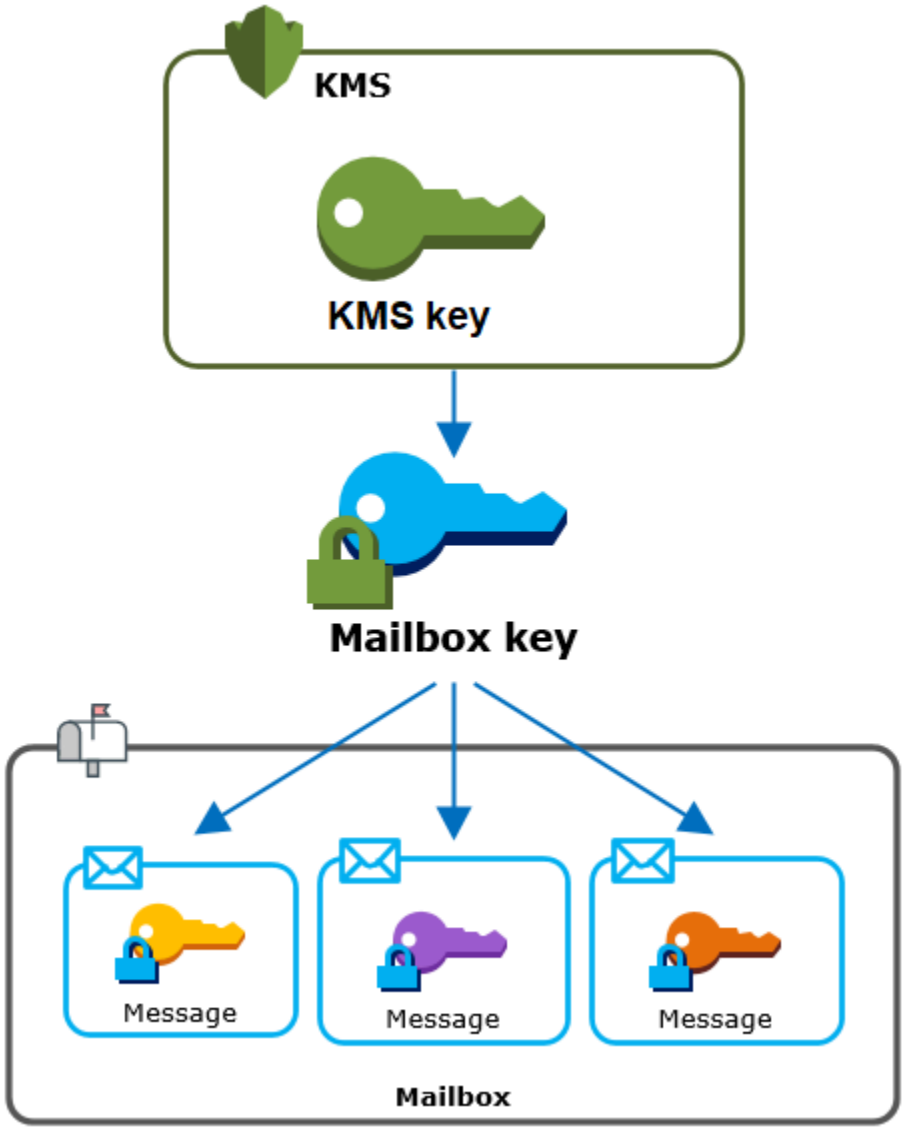
Amazon bietet WorkMail auch die Möglichkeit, Benutzern das Senden signierter oder verschlüsselter E-Mails zu ermöglichen. Diese Verschlüsselungsfunktion verwendet nicht AWS KMS .

Amazon- WorkMail Verschlüsselung

In Amazon kann WorkMailjede Organisation mehrere Postfächer enthalten, eines für jeden Benutzer in der Organisation. Alle Nachrichten, einschließlich E-Mail und Kalender, werden im Postfach des Benutzers gespeichert.

Um den Inhalt der Postfächer in Ihren Amazon- WorkMail Organisationen zu schützen, WorkMail verschlüsselt Amazon alle Postfachnachrichten, bevor sie auf die Festplatte geschrieben werden. Keine vom Kunden bereitgestellten Informationen werden als Klartext gespeichert.

Jede Nachricht wird mit einem eindeutigen Datenverschlüsselungsschlüssel verschlüsselt. Der Nachrichtenschlüssel wird durch einen Postfach-Schlüssel geschützt. Dabei handelt es sich um einen eindeutigen Verschlüsselungsschlüssel, der nur für das Postfach verwendet wird. Der Postfach-Schlüssel ist mit einem AWS KMS key für die Organisation verschlüsselt, der AWS KMS niemals unverschlüsselt verlässt. Das folgende Diagramm zeigt die Beziehung zwischen den verschlüsselten Nachrichten, den verschlüsselten Nachrichtenschlüsseln, dem verschlüsselten Postfach-Schlüssel und dem KMS-Schlüssel für die Organisation in AWS KMS auf.



Ein KMS-Schlüssel für die Organisation

Wenn Sie eine Amazon- WorkMail Organisation erstellen, können Sie eine AWS KMS keyfür die Organisation auswählen. Dieser KMS-Schlüssel schützt alle Postfach-Schlüssel in dieser Organisation.

Wenn Sie das Quick Setup-Verfahren verwenden, um Ihre Organisation zu erstellen, WorkMail verwendet Amazon die Von AWS verwalteter Schlüssel für Amazon WorkMail (`aws/workmail`) in Ihrem AWS-Konto. Wenn Sie die Standardeinrichtung verwenden, können Sie die Von AWS verwalteter Schlüssel für Amazon WorkMail oder einen kundenverwalteten Schlüssel auswählen, den Sie besitzen und verwalten. Sie können den gleichen KMS-Schlüssel oder einen anderen KMS-

Schlüssel für jede Ihrer Organisationen auswählen. Einmal ausgewählt kann der KMS-Schlüssel jedoch nicht mehr geändert werden.

> **Important**
>
> Amazon WorkMail unterstützt nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Daten in Amazon zu verschlüsseln WorkMail. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Um den KMS-Schlüssel für Ihre Organisation zu finden, verwenden Sie den AWS CloudTrail-Protokolleintrag, der Aufrufe an AWS KMS aufzeichnet.

Ein eindeutiger Verschlüsselungsschlüssel für jedes Postfach

Wenn Sie ein neues Postfach erstellen, WorkMail generiert Amazon einen eindeutigen symmetrischen 256-Bit Advanced Encryption Standard (AES)-Verschlüsselungsschlüssel für das Postfach, der als Postfachschlüssel bezeichnet wird, außerhalb von AWS KMS. Amazon WorkMail verwendet den Postfachschlüssel, um die Verschlüsselungsschlüssel für jede Nachricht im Postfach zu schützen.

Zum Schutz des WorkMail Postfachschlüssels ruft Amazon auf, AWS KMS um den Postfachschlüssel mit dem KMS-Schlüssel der Organisation zu verschlüsseln. Anschließend wird der verschlüsselte Postfach-Schlüssel in den Postfach-Metadaten gespeichert.

> **Note**
>
> Amazon WorkMail verwendet einen symmetrischen Postfach-Verschlüsselungsschlüssel, um Nachrichtenschlüssel zu schützen. Zuvor WorkMail schützte Amazon jedes Postfach mit einem asymmetrischen Schlüsselpaar. Der öffentliche Schlüssel wurde zum Verschlüsseln einzelner Nachrichtenschlüssel verwendet und der private Schlüssel, um sie zu entschlüsseln. Der private Postfach-Schlüssel wurde durch den KMS-Schlüssel der Organisation geschützt. Vorhandene Postfächer verwenden unter Umständen weiterhin ein asymmetrisches Postfach-Schlüsselpaar. Diese Änderung hat keine Auswirkungen auf die Sicherheit des Postfachs oder seiner Nachrichten.

Ein eindeutiger Verschlüsselungsschlüssel für jede Nachricht

Wenn dem Postfach eine Nachricht hinzugefügt wird, WorkMail generiert Amazon einen eindeutigen symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel für die Nachricht außerhalb von AWS KMS. Es verwendet diesen Nachrichtenschlüssel, um die Nachricht zu verschlüsseln. Amazon WorkMail verschlüsselt den Nachrichtenschlüssel unter dem Postfachschlüssel und speichert den verschlüsselten Nachrichtenschlüssel mit der Nachricht. Anschließend wird der Postfach-Schlüssel mit dem KMS-Schlüssel der Organisation verschlüsselt.

Erstellen eines neuen Postfachs

Wenn Amazon ein neues Postfach WorkMail erstellt, verwendet es den folgenden Prozess, um das Postfach auf das Speichern verschlüsselter Nachrichten vorzubereiten.

- Amazon WorkMail generiert einen eindeutigen symmetrischen 256-Bit-AES-Verschlüsselungsschlüssel für das Postfach außerhalb von AWS KMS.
- Amazon WorkMail ruft die AWS KMS Encrypt-Operation auf. Dabei werden der Postfach-Schlüssel und der Bezeichner des AWS KMS key für die Organisation übergeben. AWS KMS gibt einen Chiffretext des mit dem KMS-Schlüssel verschlüsselten Postfach-Schlüssels zurück.
- Amazon WorkMail speichert den verschlüsselten Postfachschlüssel mit den Postfach-Metadaten.

Verschlüsseln einer Postfach-Nachricht

Zum Verschlüsseln einer Nachricht WorkMail verwendet Amazon den folgenden Prozess.

1. Amazon WorkMail generiert einen eindeutigen symmetrischen 256-Bit-AES-Schlüssel für die Nachricht. Zum Verschlüsseln der Nachricht außerhalb von AWS KMS werden der Klartext-Nachrichtenschlüssel und der Advanced Encryption Standard (AES)-Algorithmus angewandt.
2. Um den Nachrichtenschlüssel unter dem Postfachschlüssel zu schützen, WorkMail muss Amazon den Postfachschlüssel entschlüsseln, der immer in verschlüsselter Form gespeichert ist.

   Amazon WorkMail ruft den Vorgang AWS KMS Decrypt auf und übergibt den verschlüsselten Postfachschlüssel. AWS KMS verwendet den KMS-Schlüssel für die Organisation, um den Postfachschlüssel zu entschlüsseln, und gibt den Klartext-Postfachschlüssel an Amazon zurück WorkMail.
3. Amazon WorkMail verwendet den Klartext-Postfachschlüssel und den Advanced Encryption Standard (AES)-Algorithmus, um den Nachrichtenschlüssel außerhalb von zu verschlüsselnAWS KMS.

4. Amazon WorkMail speichert den verschlüsselten Nachrichtenschlüssel in den Metadaten der verschlüsselten Nachricht, damit er zum Entschlüsseln verfügbar ist.

Entschlüsseln einer Postfach-Nachricht

Zum Entschlüsseln einer Nachricht WorkMail verwendet Amazon den folgenden Prozess.

1. Amazon WorkMail ruft die Operation AWS KMS Decrypt auf und übergibt den verschlüsselten Postfachschlüssel. AWS KMS verwendet den KMS-Schlüssel für die Organisation, um den Postfachschlüssel zu entschlüsseln, und gibt den Klartext-Postfachschlüssel an Amazon zurück WorkMail.
2. Amazon WorkMail verwendet den Klartext-Postfachschlüssel und den Advanced Encryption Standard (AES)-Algorithmus, um den verschlüsselten Nachrichtenschlüssel außerhalb von zu entschlüsselnAWS KMS.
3. Amazon WorkMail verwendet den Klartext-Nachrichtenschlüssel, um die verschlüsselte Nachricht zu entschlüsseln.

Zwischenspeichern von Postfachschlüsseln

Um die Leistung zu verbessern und Aufrufe an zu minimierenAWS KMS, WorkMail speichert Amazon jeden Klartext-Postfachschlüssel für jeden Client lokal bis zu eine Minute lang zwischen. Am Ende des Caching-Zeitraums wird der Postfach-Schlüssel entfernt. Wenn der Postfach-Schlüssel für diesen Client während des Caching-Zeitraums erforderlich ist, WorkMail kann Amazon ihn aus dem Cache abrufen, anstatt aufzurufenAWS KMS. Der Postfach-Schlüssel wird im Cache geschützt und wird nie als Klartext auf den Datenträger geschrieben.

Autorisieren der Nutzung des KMS-Schlüssels

Wenn Amazon einen AWS KMS key in kryptografischen Operationen WorkMail verwendet, handelt es im Namen des Postfachadministrators.

Um den AWS KMS key für ein Geheimnis in Ihrem Namen verwenden zu können, benötigt der Administrator die folgenden Berechtigungen. Sie können diese erforderlichen Berechtigungen in einer IAM-Richtlinie oder einer Schlüsselrichtlinie festlegen.

- `kms:Encrypt`
- `kms:Decrypt`

- `kms:CreateGrant`

Um zu erlauben, dass der KMS-Schlüssel nur für Anforderungen verwendet wird, die von Amazon stammen WorkMail, können Sie den Bedingungsschlüssel kms:ViaService mit dem `workmail.<region>.amazonaws.com` Wert verwenden.

Sie können auch die Schlüssel oder Werte im Verschlüsselungskontext als Bedingung für die Nutzung des KMS-Schlüssels in kryptografischen Operationen verwenden. Sie können beispielsweise einen Zeichenfolgen-Bedingungsoperator in einem IAM- oder Schlüsselrichtliniendokument oder eine Erteilungseinschränkung in einer Erteilung verwenden.

Schlüsselrichtlinie für den Von AWS verwalteter Schlüssel

Die Schlüsselrichtlinie für das Von AWS verwalteter Schlüssel für Amazon WorkMail erteilt Benutzern nur dann die Berechtigung, den KMS-Schlüssel für bestimmte Operationen zu verwenden, wenn Amazon die Anforderung im Namen des Benutzers WorkMail stellt. Die Schlüsselrichtlinie erlaubt es keinem Benutzer, den KMS-Schlüssel direkt zu verwenden.

Diese Schlüsselrichtlinie wird – wie die Richtlinien aller Von AWS verwaltete Schlüssel – vom Service eingerichtet. Sie können die Schlüsselrichtlinie nicht ändern, Sie können sie jedoch jederzeit anzeigen. Details hierzu finden Sie unter Anzeigen einer Schlüsselrichtlinie.

Die Richtlinienanweisungen in der Schlüsselrichtlinie haben folgende Wirkungen:

- Erlauben Sie Benutzern im Konto und in der Region, den KMS-Schlüssel für kryptografische Operationen zu verwenden und Erteilungen zu erstellen, jedoch nur, wenn die Anforderung WorkMail in ihrem Namen von Amazon stammt. Der Bedingungsschlüssel `kms:ViaService` setzt diese Beschränkung durch.
- Erteilt dem AWS-Konto die Berechtigung zum Erstellen von IAM-Richtlinien, mit denen Benutzer KMS-Schlüssel-Eigenschaften anzeigen und Erteilungen widerrufen können.

Im Folgenden finden Sie eine Schlüsselrichtlinie für ein Beispiel Von AWS verwalteter Schlüssel für Amazon WorkMail.

```
{
  "Version" : "2012-10-17",
  "Id" : "auto-workmail-1",
  "Statement" : [ {
```

```
    "Sid" : "Allow access through WorkMail for all principals in the account that are
 authorized to use WorkMail",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [ "kms:Decrypt", "kms:CreateGrant", "kms:ReEncrypt*", "kms:DescribeKey",
 "kms:Encrypt" ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "workmail.us-east-1.amazonaws.com",
        "kms:CallerAccount" : "111122223333"
      }
    }
  }, {
    "Sid" : "Allow direct access to key metadata to the account",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : [ "kms:Describe*", "kms:List*", "kms:Get*", "kms:RevokeGrant" ],
    "Resource" : "*"
  } ]
}
```

Verwenden von Erteilungen zur Autorisierung von Amazon WorkMail

Zusätzlich zu den Schlüsselrichtlinien WorkMail verwendet Amazon Erteilungen, um dem KMS-Schlüssel für jede Organisation Berechtigungen hinzuzufügen. Um die Erteilungen für den KMS-Schlüssel in Ihrem Konto anzuzeigen, verwenden Sie die -ListGrantsOperation.

Amazon WorkMail verwendet Erteilungen, um dem KMS-Schlüssel für die Organisation die folgenden Berechtigungen hinzuzufügen.

- Fügen Sie die `kms:Encrypt` Berechtigung hinzu, WorkMail damit Amazon den Postfachschlüssel verschlüsseln kann.
- Fügen Sie die `kms:Decrypt` Berechtigung hinzu, damit Amazon den KMS WorkMail -Schlüssel zum Entschlüsseln des Postfachschlüssels verwenden kann. Amazon WorkMail benötigt diese Berechtigung in einer Erteilung, da die Anforderung zum Lesen von Postfachnachrichten den Sicherheitskontext des Benutzers verwendet, der die Nachricht liest. Die Anforderung verwendet

nicht die Anmeldeinformationen des AWS-Konto. Amazon WorkMail erstellt diese Erteilung, wenn Sie einen KMS-Schlüssel für die Organisation auswählen.

Um die Erteilungen zu erstellen, WorkMail ruft Amazon CreateGrant im Namen des Benutzers auf, der die Organisation erstellt hat. Die Berechtigung zum Erstellen der Erteilung stammt aus der Schlüsselrichtlinie. Diese Richtlinie ermöglicht es Kontobenutzern, `CreateGrant` für den KMS-Schlüssel der Organisation aufzurufen, wenn Amazon die Anforderung im Namen eines autorisierten Benutzers WorkMail stellt.

Die Schlüsselrichtlinie erlaubt dem Konto-Root außerdem, die für den Von AWS verwalteter Schlüssel geltende Erteilung zu widerrufen. Wenn Sie jedoch die Erteilung widerrufen, WorkMail kann Amazon die verschlüsselten Daten in Ihren Postfächern nicht entschlüsseln.

Amazon WorkMail -Verschlüsselungskontext

Ein Verschlüsselungskontext ist eine Gruppe von Schlüssel/Wert-Paaren mit zufälligen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext kryptografisch an die verschlüsselten Daten. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Amazon WorkMail verwendet bei allen kryptografischen Operationen dasselbe AWS KMS Verschlüsselungskontextformat. Sie können eine kryptografische Operation in Prüfungsdatensätzen und -Protokollen wie AWS CloudTrail anhand des Verschlüsselungskontexts identifizieren. Dieser kann auch als Bedingung für die Autorisierung in Richtlinien und Erteilungen verwendet werden.

In seinen Encrypt- und Decrypt-Anforderungen an WorkMail verwendet Amazon einen VerschlüsselungskontextAWS KMS, wobei der Schlüssel `aws:workmail:arn` und der Wert der Amazon-Ressourcenname (ARN) der Organisation ist.

```
"aws:workmail:arn":"arn:aws:workmail:region:account ID:organization/organization ID"
```

Der folgende Verschlüsselungskontext enthält einen Beispiel-Organisations-ARN in der Region USA Ost (Ohio) (`us-east-2`).

```
"aws:workmail:arn":"arn:aws:workmail:us-east-2:111122223333:organization/
m-68755160c4cb4e29a2b2f8fb58f359d7"
```

Überwachen der Amazon- WorkMail Interaktion mit AWS KMS

Sie können AWS CloudTrail und Amazon CloudWatch Logs verwenden, um die Anforderungen zu verfolgen, die Amazon AWS KMS in Ihrem Namen an WorkMail sendet.

Encrypt

Wenn Sie ein neues Postfach erstellen, WorkMail generiert Amazon einen Postfachschlüssel und ruft auf, AWS KMS um den Postfachschlüssel zu verschlüsseln. Amazon WorkMail sendet eine Encrypt-Anforderung an AWS KMS mit dem Klartext-Postfachschlüssel und einer Kennung für den KMS-Schlüssel der Amazon- WorkMail Organisation.

Das Ereignis, das die `Encrypt`-Operation aufzeichnet, ähnelt dem folgenden Beispielereignis. Der Benutzer ist der Amazon- WorkMail Service. Zu den Parametern gehören die KMS-Schlüssel-ID (`keyId`) und der Verschlüsselungskontext für die Amazon- WorkMail Organisation. Amazon übergibt WorkMail auch den Postfachschlüssel, der jedoch nicht im CloudTrail Protokoll aufgezeichnet wird.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-19T10:01:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        },
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    },
    "responseElements": null,
    "requestID": "76e96b96-7e24-4faf-a2d6-08ded2eaf63c",
    "eventID": "d5a59c18-128a-4082-aa5b-729f7734626a",
    "readOnly": true,
    "resources": [
```

```
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "d08e60f1-097e-4a00-b7e9-10bc3872d50c"
}
```

Decrypt

Wenn Sie eine Postfachnachricht hinzufügen, anzeigen oder löschen, WorkMail fordert Amazon auf, den Postfachschlüssel AWS KMS zu entschlüsseln. Amazon WorkMail sendet eine Decrypt-Anforderung an AWS KMS mit dem verschlüsselten Postfachschlüssel und einer Kennung für den KMS-Schlüssel der Amazon WorkMail-Organisation.

Das Ereignis, das die `Decrypt`-Operation aufzeichnet, ähnelt dem folgenden Beispielereignis. Der Benutzer ist der Amazon- WorkMail Service. Zu den Parametern gehören der verschlüsselte Postfachschlüssel (als Geheimtext-Blob), der nicht im Protokoll aufgezeichnet wird, und der Verschlüsselungskontext für die Amazon- WorkMail Organisation. AWS KMS leitet die ID des KMS-Schlüssels aus dem Geheimtext ab.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-20T11:51:10Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        }
```

```
    },
    "responseElements": null,
    "requestID": "4a32dda1-34d9-4100-9718-674b8e0782c9",
    "eventID": "ea9fd966-98e9-4b7b-b377-6e5a397a71de",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "241e1e5b-ff64-427a-a5b3-7949164d0214"
}
```


Wie WorkSpaces verwendet AWS KMS

Sie können verwendenWorkSpaces, um einen cloudbasierten Desktop (ein WorkSpace) für jeden Ihrer Endbenutzer bereitzustellen. Wenn Sie ein neues starten WorkSpace, können Sie die Volumes verschlüsseln und entscheiden, welche für die Verschlüsselung verwendet werden AWS KMS key soll. Sie können den Von AWS verwalteter Schlüssel für WorkSpaces (aws/workspaces) oder einen symmetrischen, vom Kunden verwalteten Schlüssel auswählen.

> **Important**
>
> WorkSpaces unterstützt nur KMS-Schlüssel mit symmetrischer Verschlüsselung. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um die Volumes in einem zu verschlüsseln WorkSpaces. Informationen zur Feststellung, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Erkennen asymmetrischer KMS-Schlüssel.

Weitere Informationen zum Erstellen von WorkSpaces mit verschlüsselten Volumes finden Sie unter Verschlüsseln eines WorkSpace im Amazon- WorkSpaces Administratorhandbuch.

Themen

- Übersicht über die WorkSpaces Verschlüsselung mit AWS KMS

- WorkSpaces Verschlüsselungskontext
- Erteilen der WorkSpaces Berechtigung zur Verwendung eines KMS-Schlüssels in Ihrem Namen

Übersicht über die WorkSpaces Verschlüsselung mit AWS KMS

Wenn Sie WorkSpaces mit verschlüsselten Volumes erstellen, WorkSpaces verwendet Amazon Elastic Block Store (Amazon EBS), um diese Volumes zu erstellen und zu verwalten. Beide Services verwenden Ihre AWS KMS key, um mit den verschlüsselten Volumes zu arbeiten. Weitere Informationen zur EBS-Volume-Verschlüsselung finden Sie in der folgenden Dokumentation:

- Wie Amazon Elastic Block Store (Amazon EBS) AWS KMS nutzt. in dieser Anleitung
- Amazon-EBS-Verschlüsselung im Amazon-EC2-Benutzerhandbuch für Windows-Instances

Wenn Sie WorkSpaces mit verschlüsselten Volumes starten, funktioniert der end-to-end Prozess wie folgt:

1. Sie geben den KMS-Schlüssel an, der für die Verschlüsselung verwendet werden soll, sowie den Benutzer und das Verzeichnis WorkSpacedes . Diese Aktion erstellt eine Erteilung, die es ermöglicht, Ihren KMS WorkSpace-Schlüssel nur dafür WorkSpaces zu verwenden, d. h. nur für das , das dem angegebenen Benutzer und Verzeichnis WorkSpace zugeordnet ist.
2. WorkSpaces erstellt ein verschlüsseltes EBS-Volume für das WorkSpace und gibt den zu verwendenden KMS-Schlüssel sowie den Benutzer und das Verzeichnis des Volumes an (die gleichen Informationen, die Sie unter angegeben habenStep 1). Diese Aktion erstellt eine Erteilung, die es Amazon EBS ermöglicht, Ihren KMS-Schlüssel nur für dieses WorkSpace und Volume zu verwenden, d. h. nur für das , das dem angegebenen Benutzer und Verzeichnis WorkSpace zugeordnet ist, und nur für das angegebene Volume.
3. Amazon EBS fordert einen Volume-Datenschlüssel an, der unter Ihrem KMS-Schlüssel verschlüsselt ist, und gibt die - `Sid` und Verzeichnis-ID des WorkSpace Benutzers sowie die Volume-ID als Verschlüsselungskontext an.
4. AWS KMS erstellt einen neuen Datenschlüssel, verschlüsselt diesen mit Ihrem KMS-Schlüssel und sendet den verschlüsselten Datenschlüssel an Amazon EBS.
5. WorkSpaces verwendet Amazon EBS, um das verschlüsselte Volume an Ihr anzuhängen WorkSpace. Amazon EBS sendet den verschlüsselten Datenschlüssel AWS KMS mit einer -`Decrypt`Anforderung an und gibt die `Sid`, seine Verzeichnis-ID und die Volume-ID des WorkSpace Benutzers an, die als Verschlüsselungskontext verwendet wird.

6. AWS KMS verwendet Ihren KMS-Schlüssel, um den Datenschlüssel zu entschlüsseln, und sendet den Klartext-Datenschlüssel an Amazon EBS.
7. Amazon EBS verwendet den Klartext-Datenschlüssel, um alle eingehenden und ausgehenden Daten vom verschlüsselten Volume zu verschlüsseln. Amazon EBS speichert den Klartext-Datenschlüssel so lange im Speicher, wie das Volume an die angefügt ist WorkSpace.
8. Amazon EBS speichert den verschlüsselten Datenschlüssel (eingegangen unter Step 4) mit den Volume-Metadaten für die zukünftige Verwendung, falls Sie den neu starten oder neu erstellen WorkSpace.
9. Wenn Sie die verwendenAWS Management Console, um eine zu entfernen WorkSpace (oder die WorkSpaces -`TerminateWorkspaces`Aktion in der API zu verwenden), WorkSpaces und Amazon EBS die Erteilungen aufheben, die es ihnen ermöglicht haben, Ihren KMS-Schlüssel für diese zu verwenden WorkSpace.

WorkSpaces Verschlüsselungskontext

WorkSpaces verwendet Ihr nicht AWS KMS key direkt für kryptografische Operationen (wie `Encrypt`, `Decrypt`, usw.), was bedeutet`GenerateDataKey`, dass WorkSpaces keine Anfragen an sendetAWS KMS, die einen Verschlüsselungskontext enthalten. Wenn Amazon EBS jedoch einen verschlüsselten Datenschlüssel für die verschlüsselten Volumes Ihres WorkSpaces (Step 3 in der Übersicht über die WorkSpaces Verschlüsselung mit AWS KMS) anfordert und eine Klartextkopie dieses Datenschlüssels (Step 5) anfordert, enthält es Verschlüsselungskontext in der Anforderung. Der Verschlüsselungskontext enthält zusätzliche authentifizierte Daten (AAD), anhand derer AWS KMS die Datenintegrität sicherstellt. Der Verschlüsselungskontext wird zudem in Ihre AWS CloudTrail-Protokolldateien geschrieben, sodass Sie jederzeit nachvollziehen können, warum ein bestimmter AWS KMS key verwendet wurde. Amazon EBS verwendet Folgendes für den Verschlüsselungskontext:

- Die `sid` des AWS Directory Service Benutzers, der dem zugeordnet ist WorkSpace
- Die Verzeichnis-ID des AWS Directory Service Verzeichnisses, das dem zugeordnet ist WorkSpace
- Die Volume-ID des verschlüsselten Volume.

Das folgende Beispiel zeigt eine JSON-Darstellung des von Amazon EBS verwendeten Verschlüsselungskontextes:

```
{
```

```
  "aws:workspaces:sid-directoryid":
 "[S-1-5-21-277731876-1789304096-451871588-1107]@[d-1234abcd01]",
  "aws:ebs:id": "vol-1234abcd"
}
```

Erteilen der WorkSpaces Berechtigung zur Verwendung eines KMS-Schlüssels in Ihrem Namen

Sie können Ihre Workspace-Daten unter dem Von AWS verwalteter Schlüssel für WorkSpaces (aws/workspaces) oder für einen vom Kunden verwalteten Schlüssel schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, müssen Sie die WorkSpaces Berechtigung zur Verwendung des KMS-Schlüssels im Namen der WorkSpaces Administratoren in Ihrem Konto erteilen. Der Von AWS verwalteter Schlüssel für WorkSpaces verfügt standardmäßig über die erforderlichen Berechtigungen.

Gehen Sie wie folgt vor WorkSpaces, um Ihren vom Kunden verwalteten Schlüssel für die Verwendung mit vorzubereiten.

1. Hinzufügen der WorkSpaces Administratoren zur Liste der Schlüsselbenutzer in der Schlüsselrichtlinie des KMS-Schlüssels
2. Erteilen zusätzlicher Berechtigungen für die WorkSpaces Administratoren mit einer IAM-Richtlinie

WorkSpaces -Administratoren benötigen auch die Berechtigung, zu verwenden WorkSpaces. Weitere Informationen zu diesen Berechtigungen finden Sie unter Steuern des Zugriffs auf -WorkSpacesRessourcen im Amazon- WorkSpaces Administratorhandbuch.

Teil 1: Hinzufügen von WorkSpaces Administratoren zu den Schlüsselbenutzern eines KMS-Schlüssels

Um WorkSpaces Administratoren die erforderlichen Berechtigungen zu erteilen, können Sie die AWS Management Console oder die AWS KMS-API verwenden.

So fügen Sie WorkSpaces Administratoren als Schlüsselbenutzer für einen KMS-Schlüssel hinzu (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Key Management Service (AWS KMS)-Konsole unter https://console.aws.amazon.com/kms.

2. Wenn Sie die AWS-Region ändern möchten, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie die Schlüssel-ID oder den Alias Ihres bevorzugten kundenverwalteten Schlüssels aus.
5. Wählen Sie die Registerkarte Key policy (Schlüsselrichtlinie). Unter Key users (Schlüsselbenutzer), wählen Sie Add (Hinzufügen) aus.
6. Wählen Sie in der Liste der IAM-Benutzer und -Rollen die Benutzer und Rollen aus, die Ihren WorkSpaces Administratoren entsprechen, und wählen Sie dann Anfügen aus.

So fügen Sie WorkSpaces Administratoren als Schlüsselbenutzer für einen KMS-Schlüssel hinzu (AWS KMS-API)

1. Verwenden Sie die -GetKeyPolicyOperation, um die vorhandene Schlüsselrichtlinie abzurufen, und speichern Sie dann das Richtliniendokument in einer -Datei.
2. Öffnen Sie die Richtlinien in Ihrem bevorzugten Texteditor. Fügen Sie die IAM-Benutzer und -Rollen, die Ihren WorkSpaces Administratoren entsprechen, zu den Richtlinienanweisungen hinzu, die Schlüsselbenutzern die Berechtigung erteilen. Speichern Sie dann die Datei.
3. Verwenden Sie die -PutKeyPolicyOperation, um die Schlüsselrichtlinie auf den KMS-Schlüssel anzuwenden.

Teil 2: Erteilen zusätzlicher Berechtigungen für WorkSpaces Administratoren

Wenn Sie einen vom Kunden verwalteten Schlüssel zum Schutz Ihrer WorkSpaces Daten verwenden, benötigen WorkSpaces Administratoren zusätzlich zu den Berechtigungen im Abschnitt Schlüsselbenutzer der Standardschlüsselrichtlinie die Berechtigung, Erteilungen für den KMS-Schlüssel zu erstellen. Wenn sie die verwenden, AWS Management Console um WorkSpaces mit verschlüsselten Volumes zu erstellen, benötigen WorkSpaces Administratoren außerdem die Berechtigung, Aliase und Listenschlüssel aufzulisten. Weitere Informationen zum Erstellen von IAM-Benutzer-Richtlinien finden Sie unter Verwaltete Richtlinien und Inline-Richtlinien im IAM-Benutzerhandbuch.

Um Ihren WorkSpaces Administratoren diese Berechtigungen zu erteilen, verwenden Sie eine IAM-Richtlinie. Fügen Sie der IAM-Richtlinie für jeden WorkSpaces Administrator eine -Richtlinienanweisung ähnlich dem folgenden Beispiel hinzu. Ersetzen Sie den Beispiel-KMS-Schlüssel-ARN (`arn:aws:kms:us-`

west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab) durch einen gültigen. Wenn Ihre WorkSpaces Administratoren nur die WorkSpaces -API (nicht die Konsole) verwenden, können Sie die zweite Richtlinienanweisung mit den `"kms:ListKeys"` Berechtigungen `"kms:ListAliases"` und weglassen.

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

Programmieren der AWS KMS-API

Sie können die AWS KMS-API verwenden, um KMS-Schlüssel und spezielle Funktionen, wie benutzerdefinierte Schlüsselspeicher zu erstellen und zu verwalten. Die KMS-Schlüssel können Sie in kryptografischen Operationen verwenden. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz.

Der Beispiel-Code in den folgenden Themen zeigt, wie Sie mithilfe der AWS-SDKs die AWS KMS-API aufrufen.

Informationen zur Verwendung der AWS KMS-Konsole zum Ausführen einiger dieser Aufgaben finden Sie unter Schlüssel verwalten.

Themen

Erstellen eines Clients

Um die AWS SDK for Java, die AWS SDK for .NET, die AWS SDK for Python (Boto3), die AWS SDK for PHP, AWS SDK for Rubydie oder das SDK AWS für JavaScript in Node.js zum Schreiben von Code zu verwenden, der die AWS Key Management Service (AWS KMS)-API verwendet, erstellen Sie zunächst einen -AWS KMSClient.

Das erstellte Clientobjekt wird im Beispielcode in den folgenden Themen verwendet.

Java

Verwenden Sie zum Erstellen eines AWS KMS-Clients in Java den Client-Generator.

```
AWSKMS kmsClient = AWSKMSClientBuilder.standard().build();
```

Weitere Informationen zum Java Client-Builder finden Sie in den folgenden Ressourcen:

- Gut funktionierende Client Builder im AWS-Entwickler-Blog
- Erstellen von Service-Clients im AWS SDK for Java-Entwicklerhandbuch
- AWSKMSClientBuilder in der AWS SDK for Java-API-Referenz

C#

```
AmazonKeyManagementServiceClient kmsClient = new AmazonKeyManagementServiceClient();
```

Python

```
kms_client = boto3.client('kms')
```

Ruby

```
require 'aws-sdk-kms'  # in v2: require 'aws-sdk'

kmsClient = Aws::KMS::Client.new
```

PHP

Verwenden Sie zum Erstellen eines AWS KMS-Clients in PHP ein AWS KMS-Client-Objekt und geben Sie die Version `2014-11-01` an. Weitere Informationen finden Sie unter KMSClient-Klasse in der AWS SDK for PHP-API-Referenz.

```
// Create a KMSClient
$KmsClient = new Aws\Kms\KmsClient([
    'profile' => 'default',
    'version' => '2014-11-01',
    'region'  => 'us-east-1'
]);
```

Node.js

```
const kmsClient = new AWS.KMS();
```

Arbeiten mit Schlüsseln

Die Beispiele in diesem Thema nutzen die AWS KMS-API zum Erstellen, Anzeigen, Aktivieren und Deaktivieren von AWS KMS AWS KMS keys und zum Generieren von Datenschlüsseln.

Themen

Erstellen eines KMS-Schlüssels

Verwenden Sie die -CreateKeyOperation, um einen AWS KMS key (KMS-Schlüssel) zu erstellen. In den Beispielen in diesem Abschnitt wird ein KMS-Schlüssel mit symmetrischer Verschlüsselung erstellt. Der in diesen Beispielen verwendete `Description`-Parameter ist optional.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Hilfestellung beim Erstellen von KMS-Schlüsseln in der AWS KMS-Konsole finden Sie unter Erstellen von Schlüsseln.

Java

Weitere Informationen finden Sie in der Beschreibung der createKey-Methode in der AWS SDK for Java-API-Referenz.

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest().withDescription(desc);
CreateKeyResult result = kmsClient.createKey(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der CreateKey encrypt-Methode im AWS SDK for .NET.

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest()
{
    Description = desc
};
CreateKeyResponse response = kmsClient.CreateKey(req);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `create_key`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kms_client.create_key(
    Description=desc
)
```

Ruby

Weitere Informationen finden Sie unter der `create_key`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kmsClient.create_key({
  description: desc
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der CreateKeyencrypt-Methode im AWS SDK for PHP.

```
// Create a KMS key
//
$desc = "Key for protecting critical data";

$result = $KmsClient->createKey([
    'Description' => $desc
]);
```

Node.js

Weitere Informationen finden Sie in der createKey-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Create a KMS key
//
const Description = 'Key for protecting critical data';

kmsClient.createKey({ Description }, (err, data) => {
  ...
});
```

PowerShell

Um einen KMS-Schlüssel in zu erstellen PowerShell, verwenden Sie das Cmdlet New-KmsKey.

```
# Create a KMS key

$desc = 'Key for protecting critical data'
New-KmsKey -Description $desc
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Generieren eines Datenschlüssels

Um einen symmetrischen Datenschlüssel zu generieren, verwenden Sie die -GenerateDataKeyOperation. Diese Operation gibt einen Klartext-Datenschlüssel und eine Kopie dieses Datenschlüssels zurück, der mit dem von Ihnen angegebenen KMS-Schlüssel mit symmetrischer Verschlüsselung verschlüsselt wurde. Sie müssen entweder eine `KeySpec` oder `NumberOfBytes` (aber nicht beide) in jedem Befehl angeben.

Hilfe zur Verwendung des Datenschlüssels für die Datenverschlüsselung finden Sie im AWS Encryption SDK. Sie können den Datenschlüssel auch in HMAC-Operationen verwenden.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der -generateDataKey Methode in der APIAWS SDK for Java-Referenz zu .

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest();
dataKeyRequest.setKeyId(keyId);
dataKeyRequest.setKeySpec("AES_256");

GenerateDataKeyResult dataKeyResult = kmsClient.generateDataKey(dataKeyRequest);

ByteBuffer plaintextKey = dataKeyResult.getPlaintext();

ByteBuffer encryptedKey = dataKeyResult.getCiphertextBlob();
```

C#

Weitere Informationen finden Sie in der Beschreibung der GenerateDataKey encrypt-Methode im AWS SDK for .NET.

```
// Generate a data key
//
```

```
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest()
{
    KeyId = keyId,
    KeySpec = DataKeySpec.AES_256
};

GenerateDataKeyResponse dataKeyResponse = kmsClient.GenerateDataKey(dataKeyRequest);

MemoryStream plaintextKey = dataKeyResponse.Plaintext;

MemoryStream encryptedKey = dataKeyResponse.CiphertextBlob;
```

Python

Weitere Informationen finden Sie in der Beschreibung der `generate_data_keyencrypt`-Methode im AWS SDK for Python (Boto3).

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.generate_data_key(
    KeyId=key_id,
    KeySpec='AES_256'
)

plaintext_key = response['Plaintext']

encrypted_key = response['CiphertextBlob']
```

Ruby

Weitere Informationen finden Sie unter der `generate_data_key`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
```

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

response = kmsClient.generate_data_key({
  key_id: key_id,
  key_spec: 'AES_256'
})

plaintext_key = response.plaintext

encrypted_key = response.ciphertext_blob
```

PHP

Weitere Informationen finden Sie in der Beschreibung der GenerateDataKeyencrypt-Methode im AWS SDK for PHP.

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$keySpec = 'AES_256';

$result = $KmsClient->generateDataKey([
    'KeyId' => $keyId,
    'KeySpec' => $keySpec,
]);

$plaintextKey = $result['Plaintext'];

$encryptedKey = $result['CiphertextBlob'];
```

Node.js

Weitere Informationen finden Sie unter der -generateDataKey Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
```

```
const KeySpec = 'AES_256';
kmsClient.generateDataKey({ KeyId, KeySpec }, (err, data) => {
  if (err) console.log(err, err.stack);
  else {
    const { CiphertextBlob, Plaintext } = data;
      ...
  }
});
```

PowerShell

Verwenden Sie das New-KMS-DataKeyCmdlet, um einen symmetrischen Datenschlüssel zu generieren.

In der Ausgabe sind der Klartextschlüssel (in der -`Plaintext`Eigenschaft) und der verschlüsselte Schlüssel (in der -`CiphertextBlob`Eigenschaft) MemoryStream Objekte. Um sie in Zeichenfolgen zu konvertieren, verwenden Sie die Methoden der `MemoryStream` Klasse oder ein Cmdlet oder eine Funktion, die `MemoryStream` Objekte in Zeichenfolgen konvertiert, z. B. die Funktionen ConvertFrom-MemoryStream und ConvertFrom-Base64 im Modul Konvertieren.

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$keySpec = 'AES_256'

$response = New-KmsDataKey -KeyId $keyId -KeySpec $keySpec
$plaintextKey = $response.Plaintext
$encryptedKey = $response.CiphertextBlob
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Anzeigen eines AWS KMS key

Um detaillierte Informationen zu einem AWS KMS key, einschließlich des KMS-Schlüssel-ARN und des Schlüsselstatus, zu erhalten, verwenden Sie die -DescribeKeyOperation.

Mit `DescribeKey` können keine Aliasnamen abgerufen werden. Um Aliase abzurufen, verwenden Sie die -ListAliasesOperation. Beispiele finden Sie unter Arbeiten mit Aliasen.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Hilfestellung beim Anzeigen von KMS-Schlüsseln in der AWS KMS-Konsole finden Sie unter Anzeigen von Schlüsseln.

Java

Weitere Informationen finden Sie in der Beschreibung der describeKey-Methode in der AWS SDK for Java-API-Referenz.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest req = new DescribeKeyRequest().withKeyId(keyId);
DescribeKeyResult result = kmsClient.describeKey(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der DescribeKey encrypt-Methode im AWS SDK for .NET.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest describeKeyRequest = new DescribeKeyRequest()
{
    KeyId = keyId
};

DescribeKeyResponse describeKeyResponse = kmsClient.DescribeKey(describeKeyRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der describe_keyencrypt-Methode im AWS SDK for Python (Boto3).

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.describe_key(
    KeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der describe_key-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.describe_key({
  key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der DescribeKeyencrypt-Methode im AWS SDK for PHP.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->describeKey([
```

```
    'KeyId' => $keyId,
]);
```

Node.js

Weitere Informationen finden Sie in der describeKey-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.describeKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um detaillierte Informationen zu einem KMS-Schlüssel zu erhalten, verwenden Sie das Get---KmsKeyCmdlet.

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Get-KmsKey -KeyId $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Abruf von Schlüssel-IDs und Schlüssel-ARNs von KMS-Schlüsseln

Um die Schlüssel-IDs und Schlüssel-ARNs des abzurufenAWS KMS keys, verwenden Sie die -ListKeysOperation. In diesen Beispielen wird der optionale `Limit`-Parameter verwendet, der die maximale Anzahl von KMS-Schlüsseln festlegt, die bei jedem Aufruf zurückgegeben werden. Informationen zur Identifizierung eines KMS-Schlüssels in einer AWS KMS-Operation finden Sie unter Schlüsselkennungen (KeyId).

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Hilfestellung beim Suchen von Schlüssel-IDs und Schlüssel-ARNs in der AWS KMS-Konsole finden Sie unter Finden der Schlüssel-ID und des Schlüssel-ARN.

Java

Weitere Informationen finden Sie in der Beschreibung der listKeys-Methode in der AWS SDK for Java-API-Referenz.

```
// List KMS keys in this account
//
Integer limit = 10;

ListKeysRequest req = new ListKeysRequest().withLimit(limit);
ListKeysResult result = kmsClient.listKeys(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListKeys encrypt-Methode im AWS SDK for .NET.

```
// List KMS keys in this account
//
int limit = 10;

ListKeysRequest listKeysRequest = new ListKeysRequest()
{
    Limit = limit
};
ListKeysResponse listKeysResponse = kmsClient.ListKeys(listKeysRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `list_keys`encrypt-Methode im AWS SDK for Python (Boto3).

```
# List KMS keys in this account

response = kms_client.list_keys(
    Limit=10
```

```
)
```

Ruby

Weitere Informationen finden Sie unter der `list_keys`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# List KMS keys in this account

response = kmsClient.list_keys({
  limit: 10
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der ListKeys encrypt-Methode im AWS SDK for PHP.

```
// List KMS keys in this account
//
$limit = 10;

$result = $KmsClient->listKeys([
    'Limit' => $limit,
]);
```

Node.js

Weitere Informationen finden Sie in der listKeys-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// List KMS keys in this account
//
const Limit = 10;
kmsClient.listKeys({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Um die Schlüssel-ID und den Schlüssel-ARN aller KMS-Schlüssel im Konto und in der Region abzurufen, verwenden Sie das Get--KmsKeyList-Cmdlet.

Um die maximale Anzahl der Ausgabeobjekte zu begrenzen, verwendet dieses Beispiel das Select-Object-cmdlet anstelle des `Limit`-Parameters, der in Listen-cmdlets veraltet ist. Hilfe zum Paginieren der Ausgabe in AWS Tools for PowerShell finden Sie unter Ausgabepaginierung mit AWS Tools for PowerShell.

```
# List KMS keys in this account

$limit = 10
Get-KmsKeyList | Select-Object -First $limit
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Aktivieren von AWS KMS keys

Um eine deaktivierte zu aktivierenAWS KMS key, verwenden Sie die -EnableKeyOperation.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Hilfestellung beim Aktivieren und Deaktivieren von KMS-Schlüssel in der AWS KMS-Konsole finden Sie unter Aktivieren und Deaktivieren von Schlüsseln.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der enableKey-Methode in der AWS SDK for Java-API-Referenz.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest req = new EnableKeyRequest().withKeyId(keyId);
kmsClient.enableKey(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der EnableKey encrypt-Methode im AWS SDK for .NET.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest enableKeyRequest = new EnableKeyRequest()
{
    KeyId = keyId
};
kmsClient.EnableKey(enableKeyRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `enable_key`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.enable_key(
    KeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der `enable_key`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
response = kmsClient.enable_key({
  key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der EnableKeyencrypt-Methode im AWS SDK for PHP.

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->enableKey([
    'KeyId' => $keyId,
]);
```

Node.js

Weitere Informationen finden Sie unter der enableKey-Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.enableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um einen KMS-Schlüssel zu aktivieren, verwenden Sie das Cmdlet Enable-KmsKey.

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
Enable-KmsKey -KeyId $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Deaktivieren von AWS KMS key

Um einen KMS-Schlüssel zu deaktivieren, verwenden Sie die -DisableKeyOperation. Das Deaktivieren eines KMS-Schlüssels verhindert, dass er in kryptografischen Operationen verwendet wird.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Hilfestellung beim Aktivieren und Deaktivieren von KMS-Schlüsseln in der AWS KMS-Konsole finden Sie unter Aktivieren und Deaktivieren von Schlüsseln.

Java

Weitere Informationen finden Sie in der Beschreibung der disableKey-Methode in der AWS SDK for Java-API-Referenz.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DisableKeyRequest req = new DisableKeyRequest().withKeyId(keyId);
kmsClient.disableKey(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der DisableKey encrypt-Methode im AWS SDK for .NET.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
```

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

DisableKeyRequest disableKeyRequest = new DisableKeyRequest()
{
    KeyId = keyId
};
kmsClient.DisableKey(disableKeyRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `disable_keyencrypt`-Methode im AWS SDK for Python (Boto3).

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.disable_key(
    KeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der `disable_key`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.disable_key({
  key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der DisableKeyencrypt-Methode im AWS SDK for PHP.

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->disableKey([
    'KeyId' => $keyId,
]);
```

Node.js

Weitere Informationen finden Sie unter der disableKey-Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.disableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um einen KMS-Schlüssel zu deaktivieren, verwenden Sie das Cmdlet Disable-KmsKey cmdlet.

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Disable-KmsKey -KeyId $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Arbeiten mit Aliasen

Die Beispiele in diesem Thema verwenden die AWS KMS-API zum Erstellen, Anzeigen, Aktualisieren und Löschen von Aliasen. Weitere Informationen zu Aliasen finden Sie unter the section called “Verwenden von Aliassen”.

Themen

Erstellen eines Alias

Wenn Sie einen AWS KMS key in der AWS Management Console erstellen, müssen Sie einen Alias dafür erstellen. Die CreateKey Operation, die einen KMS-Schlüssel erstellt, erstellt jedoch keinen Alias.

Um einen Alias zu erstellen, verwenden Sie die -CreateAliasOperation. Ein Alias muss im Konto und der Region eindeutig sein. Sie können keinen Alias erstellen, der mit `aws/` beginnt. Die `aws/`-Präfix wird von Amazon Web Services für Von AWS verwaltete Schlüssel reserviert.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der createAlias-Methode in der AWS SDK for Java-API-Referenz.

```
// Create an alias for a KMS key
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest req = new
 CreateAliasRequest().withAliasName(aliasName).withTargetKeyId(targetKeyId);
```

```
kmsClient.createAlias(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der CreateAlias encrypt-Methode im AWS SDK for .NET.

```
// Create an alias for a KMS key
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest createAliasRequest = new CreateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};
kmsClient.CreateAlias(createAliasRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `create_aliasencrypt-Methode` im AWS SDK for Python (Boto3).

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.create_alias(
    AliasName=alias_name,
    TargetKeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der `create_alias`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.create_alias({
  alias_name: alias_name,
  target_key_id: target_key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der CreateAlias encrypt-Methode im AWS SDK for PHP.

```
// Create an alias for a KMS key
//
$aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';


$result = $KmsClient->createAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
]);
```

Node.js

Weitere Informationen finden Sie in der createAlias-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Create an alias for a KMS key
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
```

```
kmsClient.createAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie zum Erstellen eines Alias das Cmdlet New-KMSAlias. Bei dem Aliasnamen wird zwischen Groß- und Kleinschreibung unterschieden.

```
# Create an alias for a KMS key

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$targetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

New-KMSAlias -TargetKeyId $targetKeyId -AliasName $aliasName
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Auflisten von Aliasen

Um Aliase im Konto und in der Region aufzulisten, verwenden Sie die -ListAliasesOperation.

Standardmäßig gibt der Befehl ListAliases alle Aliase innerhalb des Kontos und der Region zurück. Darunter fallen auch Aliase, die Sie mit Ihren benutzerverwalteten Schlüssel erstellt und zugeordnet haben, und Aliase, die AWS erstellt und Ihren Von AWS verwaltete Schlüssel zugeordnet hat. Die Antwort kann auch Aliase ohne das Feld `TargetKeyId` enthalten. Dies sind vordefinierte Aliase, die von AWS erstellt, aber noch keinem KMS-Schlüssel zugeordnet wurden.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der listAlias-Methode in der AWS SDK for Java-API-Referenz.

```
// List the aliases in this AWS-Konto
```

```
//
Integer limit = 10;

ListAliasesRequest req = new ListAliasesRequest().withLimit(limit);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListAliases encrypt-Methode im AWS SDK for .NET.

```
// List the aliases in this AWS-Konto
//
int limit = 10;

ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    Limit = limit
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `list_aliasesencrypt-Methode` im AWS SDK for Python (Boto3).

```
# List the aliases in this AWS-Konto

response = kms_client.list_aliases(
    Limit=10
)
```

Ruby

Weitere Informationen finden Sie unter der `list_aliases`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# List the aliases in this AWS-Konto

response = kmsClient.list_aliases({
  limit: 10
```

```
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der Methode List Aliases in der AWS SDK for PHP.

```
// List the aliases in this AWS-Konto
//
$limit = 10;

$result = $KmsClient->listAliases([
    'Limit' => $limit,
]);
```

Node.js

Weitere Informationen finden Sie in der listAliases-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// List the aliases in this AWS-Konto
//
const Limit = 10;
kmsClient.listAliases({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Get-KMS-AliasListCmdlet, um die Aliase im Konto und in der Region aufzulisten.

Um die maximale Anzahl der Ausgabeobjekte zu begrenzen, verwendet dieses Beispiel das Select-Object-cmdlet anstelle des `Limit`-Parameters, der in Listen-cmdlets veraltet ist. Hilfe zum Paginieren der Ausgabe in AWS Tools for PowerShell finden Sie unter Ausgabepaginierung mit AWS Tools for PowerShell.

```
# List the aliases in this AWS-Konto
$limit = 10
```

```
$result = Get-KMSAliasList | Select-Object -First $limit
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Wenn Sie nur die Aliase auflisten möchten, die einem bestimmten KMS-Schlüssel zugeordnet sind, verwenden Sie den `KeyId`-Parameter. Sein Wert kann die Schlüssel-ID oder der Schlüssel-ARN eines beliebigen KMS-Schlüssels in der Region sein. Sie können keinen Aliasnamen oder Alias-ARN angeben.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der listAliases-Methode in der AWS SDK for Java-API-Referenz.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListAliasesRequest req = new ListAliasesRequest().withKeyId(keyId);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListAliases encrypt-Methode im AWS SDK for .NET.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    KeyId = keyId
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `list_aliasesencrypt-Methode` im AWS SDK for Python (Boto3).

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_aliases(
    KeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der `list_aliases`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_aliases({
  key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der Methode List Aliases in der AWS SDK for PHP.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listAliases([
```

```
    'KeyId' => $keyId,
]);
```

Node.js

Weitere Informationen finden Sie in der listAliases-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.listAliases({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um die Aliase für einen KMS-Schlüssel aufzulisten, verwenden Sie den -KeyIdParameter des Get-KMS-Cmdlets. AliasList

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

$response = Get-KmsAliasList -KeyId $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Aktualisieren eines Alias

Um einen vorhandenen Alias einem anderen KMS-Schlüssel zuzuordnen, verwenden Sie die -UpdateAliasOperation.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der updateAlias-Methode in der AWS SDK for Java-API-Referenz.

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest req = new UpdateAliasRequest()
     .withAliasName(aliasName)
     .withTargetKeyId(targetKeyId);

kmsClient.updateAlias(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der UpdateAlias encrypt-Methode im AWS SDK for .NET.

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest updateAliasRequest = new UpdateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};

kmsClient.UpdateAlias(updateAliasRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `update_aliasencrypt-Methode` im AWS SDK for Python (Boto3).

```
# Updating an alias

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kms_client.update_alias(
    AliasName=alias_name,
    TargetKeyID=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der `update_alias`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Updating an alias

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kmsClient.update_alias({
  alias_name: alias_name,
  target_key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der UpdateAlias encrypt-Methode im AWS SDK for PHP.

```
// Updating an alias
//
$aliasName = "alias/projectKey1";

// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';
```

```
$result = $KmsClient->updateAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
]);
```

Node.js

Weitere Informationen finden Sie in der updateAlias-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Updating an alias
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';
kmsClient.updateAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie zum Ändern des KMS-Schlüssels, der einem Alias zugeordnet ist das Update-KMSAlias-cmdlet. Bei dem Aliasnamen wird zwischen Groß- und Kleinschreibung unterschieden.

Das Cmdlet `Update-KMSAlias` gibt keine Ausgabe zurück. Um zu überprüfen, ob der Befehl funktioniert hat, verwenden Sie das Get-KMSAliasList-Cmdlet.

```
# Updating an alias

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

Update-KMSAlias -AliasName $aliasName -TargetKeyID $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Löschen eines Alias

Um einen Alias zu löschen, verwenden Sie die -DeleteAliasOperation. Das Löschen eines Alias hat keine Auswirkungen auf den zugeordneten KMS-Schlüssel.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der deleteAlias-Methode in der AWS SDK for Java-API-Referenz.

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest req  = new DeleteAliasRequest().withAliasName(aliasName);
kmsClient.deleteAlias(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der DeleteAlias encrypt-Methode im AWS SDK for .NET.

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest deleteAliasRequest = new DeleteAliasRequest()
{
    AliasName = aliasName
};
kmsClient.DeleteAlias(deleteAliasRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `delete_aliasencrypt-Methode` im AWS SDK for Python (Boto3).

```
# Delete an alias for a KMS key
```

```
alias_name = 'alias/projectKey1'

response = kms_client.delete_alias(
    AliasName=alias_name
)
```

Ruby

Weitere Informationen finden Sie unter der `delete_alias`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kmsClient.delete_alias({
  alias_name: alias_name
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der DeleteAlias encrypt-Methode im AWS SDK for PHP.

```
// Delete an alias for a KMS key
//
$aliasName = "alias/projectKey1";

$result = $KmsClient->deleteAlias([
    'AliasName' => $aliasName,
]);
```

Node.js

Weitere Informationen finden Sie in der deleteAlias-Eigenschaft) im AWS -SDK für JavaScript in Node.js .

```
// Delete an alias for a KMS key
//
const AliasName = 'alias/projectKey1';
kmsClient.deleteAlias({ AliasName }, (err, data) => {
  ...
```

```
});
```

PowerShell

Verwenden Sie zum Löschen eines Alias das Cmdlet Remove-KMSAlias. Bei dem Aliasnamen wird zwischen Groß- und Kleinschreibung unterschieden.

Da dieses Cmdlet den Alias dauerhaft löscht, PowerShell werden Sie aufgefordert, den Befehl zu bestätigen. Die `ConfirmImpact` ist `High`, sodass Sie keine `ConfirmPreference` verwenden können, um diese Aufforderung zu unterdrücken. Wenn Sie die Bestätigungsaufforderung unterdrücken müssen, fügen Sie den allgemeinen `Confirm`-Parameter mit dem Wert `$false` hinzu, z. B.: `-Confirm:$false`.

Das Cmdlet `Remove-KMSAlias` gibt keine Ausgabe zurück. Verwenden Sie das Get-KMS-AliasListCmdlet, um zu überprüfen, ob der Befehl wirksam war.

```
# Delete an alias for a KMS key

$aliasName = 'alias/projectKey1'
Remove-KMSAlias -AliasName $aliasName
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Verschlüsseln und Entschlüsseln von Datenschlüsseln

In den Beispielen in diesem Thema werden die ReEncrypt Operationen Encrypt , Decrypt und in der AWS KMS API verwendet.

Diese Operationen sind auf das Ver- und Entschlüsseln von Datenschlüsseln ausgelegt. Sie verwenden einen AWS KMS keys in den Verschlüsselungsoperationen und können nicht mehr als 4 KB (4096 Bytes) Daten entgegennehmen. Obwohl Sie sie verwenden können, um kleine Datenmengen verschlüsseln, wie beispielsweise ein Passwort oder einen RSA-Schlüssel, sind sie nicht auf die Verschlüsselung von Anwendungsdaten ausgelegt.

Um Anwendungsdaten zu verschlüsseln, verwenden Sie die serverseitigen Verschlüsselungsfunktionen eines AWS-Service oder eine clientseitige Verschlüsselungsbibliothek, wie AWS Encryption SDK oder den Simple Storage Service (Amazon S3)-Verschlüsselungsclient.

Themen

Verschlüsseln eines Datenschlüssels

Die Produktion Encrypt ist auf die Verschlüsselung von Datenschlüsseln ausgelegt, wird aber nicht häufig verwendet. Die GenerateDataKeyWithoutPlaintext Operationen GenerateDataKey und geben verschlüsselte Datenschlüssel zurück. Verwenden Sie diese Methode, wenn Sie verschlüsselte Daten in eine andere Region verschieben und ihren Datenschlüssel in der neuen Region mit einem KMS-Schlüssel verschlüsseln möchten.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der encrypt-Methode in der AWS SDK for Java-API-Referenz.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
ByteBuffer plaintext = ByteBuffer.wrap(new byte[]{1,2,3,4,5,6,7,8,9,0});

EncryptRequest req = new EncryptRequest().withKeyId(keyId).withPlaintext(plaintext);
ByteBuffer ciphertext = kmsClient.encrypt(req).getCiphertextBlob();
```

C#

Weitere Informationen finden Sie in der Beschreibung der encrypt-Methode im AWS SDK for .NET.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
```

```
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
MemoryStream plaintext = new MemoryStream();
plaintext.Write(new byte[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }, 0, 10);

EncryptRequest encryptRequest = new EncryptRequest()
{
    KeyId = keyId,
    Plaintext = plaintext
};
MemoryStream ciphertext = kmsClient.Encrypt(encryptRequest).CiphertextBlob;
```

Python

Weitere Informationen finden Sie in in der Beschreibung der encrypt-Methode im AWS SDK for Python (Boto3).

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00'

response = kms_client.encrypt(
    KeyId=key_id,
    Plaintext=plaintext
)

ciphertext = response['CiphertextBlob']
```

Ruby

Weitere Informationen finden Sie unter der Instance-Methode encrypt im AWS SDK for Ruby.

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00"

response = kmsClient.encrypt({
```

```
  key_id: key_id,
  plaintext: plaintext
})

ciphertext = response.ciphertext_blob
```

PHP

Weitere Informationen finden Sie in der Beschreibung der encrypt-Methode im AWS SDK for PHP.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$message = pack('c*',1,2,3,4,5,6,7,8,9,0);

$result = $KmsClient->encrypt([
    'KeyId' => $keyId,
    'Plaintext' => $message,
]);

$ciphertext = $result['CiphertextBlob'];
```

Node.js

Weitere Informationen finden Sie unter der encrypt-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Plaintext = Buffer.from([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]);
kmsClient.encrypt({ KeyId, Plaintext }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { CiphertextBlob } = data;
      ...
  }
});
```

PowerShell

Verwenden Sie zum Verschlüsseln eines Datenschlüssels mit einem KMS-Schlüssel das cmdlet Invoke-KMSEncrypt. Es gibt den Geheimtext als `MemoryStream` (System.IO.MemoryStream)-Objekt zurück. Sie können das `MemoryStream`-Objekt als Eingabe für das Cmdlet Invoke-KMSDecrypt verwenden.

AWS KMS gibt auch Datenschlüssel als `MemoryStream`-Objekte zurück. Um in diesem Beispiel einen Klartext-Datenschlüssel zu simulieren, erstellen wir ein Byte-Array und schreiben es in ein `MemoryStream`-Objekt.

Beachten Sie, dass der `Plaintext`-Parameter von `Invoke-KMSEncrypt` ein Byte-Array (`byte[]`) verwendet; es ist kein `MemoryStream`-Objekt erforderlich. Ab AWSPowerShell Version 4.0 akzeptieren Parameter in allen AWSPowerShell Modulen, die Byte-Arrays und `MemoryStream` Objekte verwenden, Byte-Arrays, `MemoryStream` Objekte, Zeichenfolgen, Zeichenfolge-Arrays und `FileInfo` (System.IO.FileInfo) Objekte. Sie können jeden dieser Typen an `Invoke-KMSEncrypt` übergeben.

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Simulate a data key
 # Create a byte array
[byte[]] $bytes = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0

 # Create a MemoryStream
$plaintext = [System.IO.MemoryStream]::new()

 # Add the byte array to the MemoryStream
$plaintext.Write($bytes, 0, $bytes.length)

# Encrypt the simulated data key
$response = Invoke-KMSEncrypt -KeyId $keyId -Plaintext $plaintext

# Get the ciphertext from the response
$ciphertext = $response.CiphertextBlob
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Entschlüsseln eines Datenschlüssels

Zur Entschlüsselung eines Datenschlüssels verwenden Sie die Produktion Decrypt.

Die `ciphertextBlob` von Ihnen angegebene muss der Wert des `CiphertextBlob` Feldes aus einer GenerateDataKey, GenerateDataKeyWithoutPlaintextoder Encrypt-Antwort oder das `PrivateKeyCiphertextBlob` Feld aus einer GenerateDataKeyPair- oder -GenerateDataKeyPairWithoutPlaintextAntwort sein. Sie können die `Decrypt`-Produktion auch verwenden, um Daten zu entschlüsseln, die außerhalb von AWS KMS mit dem öffentlichen Schlüssel in einem asymmetrischen KMS-Schlüssel verschlüsselt werden.

Der `KeyId`-Parameter ist beim Entschlüsseln mit KMS-Schlüsseln mit symmetrischer Verschlüsselung nicht erforderlich. AWS KMS kann den KMS-Schlüssel abrufen, der zum Verschlüsseln der Daten aus den Metadaten im Verschlüsselungstext-Blob verwendet wurde. Es ist jedoch immer eine bewährte Methode, den von Ihnen verwendeten KMS-Schlüssel anzugeben. Diese Methode stellt sicher, dass Sie den beabsichtigten KMS-Schlüssel verwenden, und verhindert, dass Sie versehentlich einen Chiffretext mit einem KMS-Schlüssel entschlüsseln, dem Sie nicht vertrauen.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der decrypt-Methode in der AWS SDK for Java-API-Referenz.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ByteBuffer ciphertextBlob = Place your ciphertext here;
```

```
DecryptRequest req = new
 DecryptRequest().withCiphertextBlob(ciphertextBlob).withKeyId(keyId);
ByteBuffer plainText = kmsClient.decrypt(req).getPlaintext();
```

C#

Weitere Informationen finden Sie in der Beschreibung der Methode Decrypt im AWS SDK for .NET.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

MemoryStream ciphertextBlob = new MemoryStream();
// Write ciphertext to memory stream

DecryptRequest decryptRequest = new DecryptRequest()
{
    CiphertextBlob = ciphertextBlob,
    KeyId = keyId
};
MemoryStream plainText = kmsClient.Decrypt(decryptRequest).Plaintext;
```

Python

Weitere Informationen finden Sie in der Beschreibung der Methode decrypt im AWS SDK for Python (Boto3).

```
# Decrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
ciphertext = 'Place your ciphertext here'

response = kms_client.decrypt(
    CiphertextBlob=ciphertext,
    KeyId=key_id
)
```

```
plaintext = response['Plaintext']
```

Ruby

Weitere Informationen finden Sie in der Beschreibung der Instance-Methode decrypt im AWS SDK for Ruby.

```
# Decrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

response = kmsClient.decrypt({
  ciphertext_blob: ciphertext_packed,
  key_id: key_id
})

plaintext = response.plaintext
```

PHP

Weitere Informationen finden Sie in der Beschreibung der Methode Decrypt im AWS SDK for PHP.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$ciphertext = 'Place your cipher text blob here';

$result = $KmsClient->decrypt([
     'CiphertextBlob' => $ciphertext,
     'KeyId' => $keyId,
]);

$plaintext = $result['Plaintext'];
```

Node.js

Weitere Informationen finden Sie in der decrypt-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const CiphertextBlob = 'Place your cipher text blob here';
kmsClient.decrypt({ CiphertextBlob, KeyId }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { Plaintext } = data;
    ...
  }
});
```

PowerShell

Verwenden Sie zum Entschlüsseln eines Datenschlüssels das Cmdlet Invoke-KMSEncrypt.

Dieses Cmdlet gibt den Klartext als `MemoryStream` (System.IO.MemoryStream)-Objekt zurück. Um es in ein Byte-Array zu konvertieren, verwenden Sie Cmdlets oder Funktionen, die `MemoryStream`-Objekte in Byte-Arrays konvertieren, z. B. die Funktionen im Modul Convert.

Da in diesem Beispiel der Verschlüsselungstext verwendet wird, den ein AWS KMS-Verschlüsselungs-Cmdlet zurückgegeben hat, wird ein `MemoryStream`-Objekt für den Wert des `CiphertextBlob`-Parameters verwendet. Der `CiphertextBlob`-Parameter von `Invoke-KMSDecrypt` verwendet jedoch ein Byte-Array (`byte[]`); es ist kein `MemoryStream`-Objekt erforderlich. Ab AWSPowerShell Version 4.0 akzeptieren Parameter in allen AWSPowerShell Modulen, die Byte-Arrays und `MemoryStream` Objekte annehmen, Byte-Arrays, `MemoryStream` Objekte, Zeichenfolgen, Zeichenfolge-Arrays und `FileInfo` (System.IO.FileInfo) Objekte. Sie können jeden dieser Typen an `Invoke-KMSDecrypt` übergeben.

```
# Decrypt a data key
# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
[System.IO.MemoryStream]$ciphertext = Read-Host 'Place your cipher text blob here'

$response = Invoke-KMSDecrypt -CiphertextBlob $ciphertext -KeyId $keyId
$plaintext = $response.Plaintext
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Erneutes Verschlüsseln eines Datenschlüssels mit einem anderen AWS KMS key

Um einen verschlüsselten Datenschlüssel zu entschlüsseln und dann den Datenschlüssel sofort mit einem anderen neu zu verschlüsselnAWS KMS key, verwenden Sie die -ReEncryptOperation. Die Operationen werden vollständig serverseitig in AWS KMS ausgeführt, sodass Ihr Klartext niemals außerhalb von AWS KMS sichtbar ist.

Die `ciphertextBlob` von Ihnen angegebene muss der Wert des `CiphertextBlob` Feldes aus einer GenerateDataKey-, - GenerateDataKeyWithoutPlaintextoder Verschlüsselungsantwort oder das `PrivateKeyCiphertextBlob` Feld aus einer GenerateDataKeyPair - oder -GenerateDataKeyPairWithoutPlaintextAntwort sein. Sie können die `ReEncrypt`-Produktion auch verwenden, um Daten erneut zu verschlüsseln, die außerhalb von AWS KMS mit dem öffentlichen Schlüssel in einem asymmetrischen KMS-Schlüssel verschlüsselt werden.

Der `SourceKeyId`-Parameter ist beim erneuten Verschlüsseln mit KMS-Schlüsseln mit symmetrischer Verschlüsselung nicht erforderlich. AWS KMS kann den KMS-Schlüssel abrufen, der zum Verschlüsseln der Daten aus den Metadaten im Verschlüsselungstext-Blob verwendet wurde. Es ist jedoch immer eine bewährte Methode, den von Ihnen verwendeten KMS-Schlüssel anzugeben. Diese Methode stellt sicher, dass Sie den beabsichtigten KMS-Schlüssel verwenden, und verhindert, dass Sie versehentlich einen Chiffretext mit einem KMS-Schlüssel entschlüsseln, dem Sie nicht vertrauen.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der reEncrypt-Methode in der AWS SDK for Java-API-Referenz.

```
// Re-encrypt a data key

ByteBuffer sourceCiphertextBlob = Place your ciphertext here;

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest req = new ReEncryptRequest();
req.setCiphertextBlob(sourceCiphertextBlob);
req.setSourceKeyId(sourceKeyId);
req.setDestinationKeyId(destinationKeyId);
ByteBuffer destinationCipherTextBlob = kmsClient.reEncrypt(req).getCiphertextBlob();
```

C#

Weitere Informationen finden Sie in der Beschreibung der ReEncrypt encrypt-Methode im AWS SDK for .NET.

```
// Re-encrypt a data key

MemoryStream sourceCiphertextBlob = new MemoryStream();
// Write ciphertext to memory stream

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest reEncryptRequest = new ReEncryptRequest()
{
    CiphertextBlob = sourceCiphertextBlob,
    SourceKeyId = sourceKeyId,
    DestinationKeyId = destinationKeyId
};
MemoryStream destinationCipherTextBlob =
 kmsClient.ReEncrypt(reEncryptRequest).CiphertextBlob;
```

Python

Weitere Informationen finden Sie in der Beschreibung der re_encryptencrypt-Methode im AWS SDK for Python (Boto3).

```
# Re-encrypt a data key
ciphertext = 'Place your ciphertext here'

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kms_client.re_encrypt(
    CiphertextBlob=ciphertext,
    SourceKeyId=source_key_id,
    DestinationKeyId=destination_key_id
)

destination_ciphertext_blob = response['CiphertextBlob']
```

Ruby

Weitere Informationen finden Sie unter der re_encrypt-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Re-encrypt a data key

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kmsClient.re_encrypt({
  ciphertext_blob: ciphertext_packed,
  source_key_id: source_key_id,
  destination_key_id: destination_key_id
```

```
})

destination_ciphertext_blob = response.ciphertext_blob.unpack('H*')
```

PHP

Weitere Informationen finden Sie in der Beschreibung der ReEncryptencrypt-Methode im AWS SDK for PHP.

```
// Re-encrypt a data key

$ciphertextBlob = 'Place your ciphertext here';

// Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';

$result = $KmsClient->reEncrypt([
    'CiphertextBlob' => $ciphertextBlob,
    'SourceKeyId' => $sourceKeyId,
    'DestinationKeyId' => $destinationKeyId,
]);
```

Node.js

Weitere Informationen finden Sie in der reEncrypt-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Re-encrypt a data key
const CiphertextBlob = 'Place your cipher text blob here';
// Replace the following example key ARNs with valid key identfiers
const SourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const DestinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321';

kmsClient.reEncrypt({ CiphertextBlob, SourceKeyId, DestinationKeyId }, (err, data)
 => {
  ...
});
```

PowerShell

Um einen Geheimtext unter demselben oder einem anderen KMS-Schlüssel erneut zu verschlüsseln, verwenden Sie das Invoke-KMSReEncrypt-cmdlet.

Da in diesem Beispiel der Verschlüsselungstext verwendet wird, den ein AWS KMS-Verschlüsselungs-Cmdlet zurückgegeben hat, wird ein `MemoryStream`-Objekt für den Wert des `CiphertextBlob`-Parameters verwendet. Der `CiphertextBlob`-Parameter von `Invoke-KMSReEncrypt` verwendet jedoch ein Byte-Array (`byte[]`); es ist kein `MemoryStream`-Objekt erforderlich. Ab AWSPowerShell Version 4.0 akzeptieren Parameter in allen AWSPowerShell Modulen, die Byte-Arrays und `MemoryStream` Objekte verwenden, Byte-Arrays, `MemoryStream` Objekte, Zeichenfolgen, Zeichenfolge-Arrays und `FileInfo` (System.IO.FileInfo) Objekte. Sie können jeden dieser Typen an `Invoke-KMSReEncrypt` übergeben.

```
# Re-encrypt a data key

[System.IO.MemoryStream]$ciphertextBlob = Read-Host 'Place your cipher text blob
 here'

# Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

$response = Invoke-KMSReEncrypt -Ciphertext $ciphertextBlob -SourceKeyId
 $sourceKeyId -DestinationKeyId $destinationKeyId
$reEncryptedCiphertext = $response.CiphertextBlob
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Arbeiten mit Schlüsselrichtlinien

Die Beispiele in diesem Thema verwenden die AWS KMS-API zum Anzeigen und Ändern der Schlüsselrichtlinien von AWS KMS keys.

Weitere Informationen zum Verwenden von Schlüsselrichtlinien, IAM-Richtlinien und Erteilungen, um den Zugriff auf Ihre KMS-Schlüssel zu verwalten, finden Sie unter Authentifizierung

und Zugriffskontrolle für AWS KMS. Hilfe beim Schreiben und Formatieren eines JSON-Richtliniendokuments finden Sie in der IAM-JSON-Richtlinienreferenz im IAM-Benutzerhandbuch.

Themen

- Auflisten der Namen von Schlüsselrichtlinien
- Abrufen einer Schlüsselrichtlinie
- Einstellen einer Schlüsselrichtlinie

Auflisten der Namen von Schlüsselrichtlinien

Um die Namen der Schlüsselrichtlinien für einen abzurufenAWS KMS key, verwenden Sie die -ListKeyPoliciesOperation. default ist der einzige Schlüsselrichtlinienname, der zurückgegeben wird.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der -listKeyPolicies Methode in der APIAWS SDK for Java-Referenz zu .

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListKeyPoliciesRequest req = new ListKeyPoliciesRequest().withKeyId(keyId);
ListKeyPoliciesResult result = kmsClient.listKeyPolicies(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListKeyPolicies encrypt-Methode im AWS SDK for .NET.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
ListKeyPoliciesRequest listKeyPoliciesRequest = new ListKeyPoliciesRequest()
{
    KeyId = keyId
};
ListKeyPoliciesResponse listKeyPoliciesResponse =
 kmsClient.ListKeyPolicies(listKeyPoliciesRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der list_key_policiesencrypt-Methode im AWS SDK for Python (Boto3).

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_key_policies(
    KeyId=key_id
)
```

Ruby

Weitere Informationen finden Sie unter der list_key_policies-Instance-Methode encrypt im AWS SDK for Ruby.

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_key_policies({
  key_id: key_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der ListKeyPoliciesencrypt-Methode im AWS SDK for PHP.

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listKeyPolicies([
    'KeyId' => $keyId
]);
```

Node.js

Weitere Informationen finden Sie unter der -listKeyPolicies Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

kmsClient.listKeyPolicies({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um den Namen der Standard-Schlüsselrichtlinie aufzulisten, verwenden Sie das Get-KMSKeyPolicyList-Cmdlet.

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$response = Get-KMSKeyPolicyList -KeyId $keyId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Abrufen einer Schlüsselrichtlinie

Um die Schlüsselrichtlinie für einen abzurufenAWS KMS key, verwenden Sie die -GetKeyPolicyOperation.

GetKeyPolicy erfordert einen Richtliniennamen. default ist der einzige gültige Richtlinienname.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der -getKeyPolicy Methode in der APIAWS SDK for Java-Referenz zu .

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest req = new
 GetKeyPolicyRequest().withKeyId(keyId).withPolicyName(policyName);
GetKeyPolicyResult result = kmsClient.getKeyPolicy(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der GetKeyPolicy encrypt-Methode im AWS SDK for .NET.

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest getKeyPolicyRequest = new GetKeyPolicyRequest()
{
    KeyId = keyId,
    PolicyName = policyName
};
```

```
GetKeyPolicyResponse getKeyPolicyResponse =
 kmsClient.GetKeyPolicy(getKeyPolicyRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der get_key_policyencrypt-Methode im AWS SDK for Python (Boto3).

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'

response = kms_client.get_key_policy(
    KeyId=key_id,
    PolicyName=policy_name
)
```

Ruby

Weitere Informationen finden Sie unter der get_key_policy-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'

response = kmsClient.get_key_policy({
  key_id: key_id,
  policy_name: policy_name
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der GetKeyPolicy encrypt-Methode im AWS SDK for PHP.

```
// Get the policy for a KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->getKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName
]);
```

Node.js

Weitere Informationen finden Sie in der -getKeyPolicy Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const PolicyName = 'default';
kmsClient.getKeyPolicy({ KeyId, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Get-KMS-CmdletKeyPolicy, um die Schlüsselrichtlinie für einen KMS-Schlüssel abzurufen. Dieses Cmdlet gibt die Schlüsselrichtlinie als Zeichenfolge (System.String) zurück, die Sie in einem Write-KMSKeyPolicy (`PutKeyPolicy`)-Befehl verwenden können. Um die Richtlinien in der JSON-Zeichenfolge in `PSCustomObject` Objekte zu konvertieren, verwenden Sie das ConvertFrom-JSON-Cmdlet.

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'

$response = Get-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Einstellen einer Schlüsselrichtlinie

Um die Schlüsselrichtlinie für einen KMS-Schlüssel zu erstellen oder zu ersetzen, verwenden Sie die -PutKeyPolicyOperation.

`PutKeyPolicy` erfordert einen Richtliniennamen. default ist der einzige gültige Richtlinienname.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der -putKeyPolicy Methode in der APIAWS SDK for Java-Referenz zu .

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleRole\"," +
                "    \"Effect\": \"Allow\"," +
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
                "  }]" +
```

```
                "}";

PutKeyPolicyRequest req = new
 PutKeyPolicyRequest().withKeyId(keyId).withPolicy(policy).withPolicyName(policyName);
kmsClient.putKeyPolicy(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der PutKeyPolicy encrypt-Methode im AWS SDK for .NET.

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleUser\"," +
                "    \"Effect\": \"Allow\"," +
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
                "  }]" +
                "}";

PutKeyPolicyRequest putKeyPolicyRequest = new PutKeyPolicyRequest()
{
    KeyId = keyId,
    Policy = policy,
    PolicyName = policyName
};
kmsClient.PutKeyPolicy(putKeyPolicyRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der put_key_policyencrypt-Methode im AWS SDK for Python (Boto3).

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = """
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "Allow access for ExampleUser",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
        "Action": [
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:ReEncrypt*"
        ],
        "Resource": "*"
    }]
}"""

response = kms_client.put_key_policy(
    KeyId=key_id,
    Policy=policy,
    PolicyName=policy_name
)
```

Ruby

Weitere Informationen finden Sie unter der put_key_policy-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
```

```
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = "{" +
  "  \"Version\": \"2012-10-17\"," +
  "  \"Statement\": [{" +
  "    \"Sid\": \"Allow access for ExampleUser\"," +
  "    \"Effect\": \"Allow\"," +
  # Replace the following example user ARN with a valid one
  "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/ExampleKeyUserRole
\"}," +
  "    \"Action\": [" +
  "      \"kms:Encrypt\"," +
  "      \"kms:GenerateDataKey*\"," +
  "      \"kms:Decrypt\"," +
  "      \"kms:DescribeKey\"," +
  "      \"kms:ReEncrypt*\"" +
  "    ]," +
  "    \"Resource\": \"*\"" +
  "  }]" +
  "}"

response = kmsClient.put_key_policy({
  key_id: key_id,
  policy: policy,
  policy_name: policy_name
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der PutKeyPolicy encrypt-Methode im AWS SDK for PHP.

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->putKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName,
```

```
    'Policy' => '{
        "Version": "2012-10-17",
        "Id": "custom-policy-2016-12-07",
        "Statement": [
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:user/root" },
            "Action": [ "kms:*" ],
            "Resource": "*" },
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole" },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*" }
        ]
    } '
]);
```

Node.js

Weitere Informationen finden Sie in der putKeyPolicy Eigenschaft im AWS -SDK für JavaScript in Node.js .

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const PolicyName = 'default';
const Policy = `{
    "Version": "2012-10-17",
    "Id": "custom-policy-2016-12-07",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }
    ]
}`; // The key policy document

kmsClient.putKeyPolicy({ KeyId, Policy, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Write-KMS-CmdletKeyPolicy, um eine Schlüsselrichtlinie für einen KMS-Schlüssel festzulegen. Dieses Cmdlet gibt keine Ausgabe zurück. Verwenden Sie das Get-KMS-KeyPolicyCmdlet, um zu überprüfen, ob der Befehl wirksam war.

Der `Policy`-Parameter verwendet eine Zeichenfolge. Umschließen Sie die Zeichenfolge in einfache Anführungszeichen, um sie zu einer Literalzeichenfolge zu machen. Sie müssen keine Fortsetzungszeichen oder Escape-Zeichen in der Literalzeichenfolge verwenden.

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
```

```
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'
$policy = '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }]
    }'

Write-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName -Policy $policy
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Arbeiten mit Erteilungen

Die Beispiele in diesem Thema verwenden die AWS KMS-API zum Erstellen, Anzeigen, Aufheben und Widerrufen von Erteilungen für AWS KMS keys. Weitere Informationen zur Verwendung von Berechtigungserteilungen in AWS KMS finden Sie unter Erteilungen in AWS KMS.

Themen

Erstellen einer Erteilung

Um eine Erteilung für ein zu erstellenAWS KMS key, verwenden Sie die -CreateGrantOperation. Die Antwort enthält nur die Erteilungs-ID und das Erteilungs-Token. Um detaillierte Informationen über die Erteilung zu erhalten, verwenden Sie die -ListGrantsOperation, wie in gezeigtAnzeigen einer Erteilung.

In diesen Beispielen wird eine Erteilung erstellt, mit der Benutzer, die die `ExampleKeyUser` Rolle übernehmen können, die -GenerateDataKeyOperation für den KMS-Schlüssel aufrufen können, der durch den -`KeyId`Parameter identifiziert wird.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der createGrant-Methode in der AWS SDK for Java-API-Referenz.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey.toString();
```

```
CreateGrantRequest request = new CreateGrantRequest()
    .withKeyId(keyId)
    .withGranteePrincipal(granteePrincipal)
    .withOperations(operation);

CreateGrantResult result = kmsClient.createGrant(request);
```

C#

Weitere Informationen finden Sie in der Beschreibung der CreateGrant encrypt-Methode im AWS SDK for .NET.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey;

CreateGrantRequest createGrantRequest = new CreateGrantRequest()
{
    KeyId = keyId,
    GranteePrincipal = granteePrincipal,
    Operations = new List<string>() { operation }
};

CreateGrantResponse createGrantResult = kmsClient.CreateGrant(createGrantRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `create_grant`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']
```

```
response = kms_client.create_grant(
    KeyId=key_id,
    GranteePrincipal=grantee_principal,
    Operations=operation
)
```

Ruby

Weitere Informationen finden Sie unter der `create_grant`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']

response = kmsClient.create_grant({
  key_id: key_id,
  grantee_principal: grantee_principal,
  operations: operation
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der CreateGrantencrypt-Methode im AWS SDK for PHP.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
$operation = ['GenerateDataKey']

$result = $KmsClient->createGrant([
    'GranteePrincipal' => $granteePrincipal,
    'KeyId' => $keyId,
    'Operations' => $operation
```

```
]);
```

Node.js

Weitere Informationen finden Sie in der createGrant-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const GranteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser';
const Operations: ["GenerateDataKey"];
kmsClient.createGrant({ KeyId, GranteePrincipal, Operations }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie zum Erstellen einer Erteilung das Cmdlet New-KMSGrant.

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$granteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
$operation = 'GenerateDataKey'

$response = New-KMSGrant -GranteePrincipal $granteePrincipal -KeyId $keyId -
Operation $operation
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Anzeigen einer Erteilung

Um detaillierte Informationen zu den Erteilungen für einen KMS-Schlüssel zu erhalten, verwenden Sie die -ListGrantsOperation.

> **Note**
>
> Das `GranteePrincipal`-Feld in der `ListGrants`-Antwort enthält normalerweise den Berechtigungsprinzipal der Genehmigung. Wenn der Empfänger-Prinzipal in der Erteilung jedoch ein AWS-Service ist, enthält das `GranteePrincipal`-Feld den Service-Prinzipal, der mehrere verschiedene Empfänger-Prinzipale repräsentieren kann.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

In diesen Beispielen wird der optionale `Limits`-Parameter verwendet, der bestimmt, wie viele Erteilungen die Produktion zurückgibt.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der listGrants-Methode in der AWS SDK for Java-API-Referenz.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
Integer limit = 10;

ListGrantsRequest req = new ListGrantsRequest().withKeyId(keyId).withLimit(limit);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListGrants encrypt-Methode im AWS SDK for .NET.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
int limit = 10;
```

```
ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    Limit = limit
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `list_grants`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_grants(
    KeyId=key_id,
    Limit=10
)
```

Ruby

Weitere Informationen finden Sie unter der `list_grants`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_grants({
  key_id: key_id,
  limit: 10
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der ListGrantsencrypt-Methode im AWS SDK for PHP.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$limit = 10;

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'Limit' => $limit,
]);
```

Node.js

Weitere Informationen finden Sie in der listGrants-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Limit = 10;
kmsClient.listGrants({ KeyId, Limit }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Get-KMS-Cmdlet, um die Details aller AWS KMS Erteilungen für einen KMS-Schlüssel anzuzeigen. GrantList

Um die maximale Anzahl der Ausgabeobjekte zu begrenzen, verwendet dieses Beispiel das Select-Object-cmdlet anstelle des `Limit`-Parameters, der in Listen-cmdlets veraltet ist. Hilfe zum Paginieren der Ausgabe in AWS Tools for PowerShell finden Sie unter Ausgabepaginierung mit AWS Tools for PowerShell.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
$limit = 10

$response = Get-KMSGrantList -KeyId $keyId | Select-Object -First $limit
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Sie müssen den KMS-Schlüssel in jeder `ListGrants`-Produktionen angeben. Sie können die Erteilungs-Liste jedoch weiter filtern, indem Sie die Erteilungs-ID oder einen Empfänger-Prinzipal angeben. Die folgenden Beispiele erhalten nur die Erteilungen für einen KMS-Schlüssel, bei dem die `test-engineer`-Rolle der Empfänger-Prinzipal ist.

Java

Weitere Informationen zur Java-Implementierung finden Sie in der Beschreibung der listGrants-Methode in der AWS SDK for Java-API-Referenz.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String grantee = "arn:aws:iam::111122223333:role/test-engineer";

ListGrantsRequest req = new
 ListGrantsRequest().withKeyId(keyId).withGranteePrincipal(grantee);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der ListGrants encrypt-Methode im AWS SDK for .NET.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String grantee = "arn:aws:iam::111122223333:role/test-engineer";
```

```
ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    GranteePrincipal = grantee
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `list_grants`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kms_client.list_grants(
    KeyId=key_id,
    GranteePrincipal=grantee
)
```

Ruby

Weitere Informationen finden Sie unter der `list_grants`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kmsClient.list_grants({
  key_id: keyId,
  grantee_principal: grantee
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der ListGrantsencrypt-Methode im AWS SDK for PHP.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$grantee = 'arn:aws:iam::111122223333:role/test-engineer';

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'GranteePrincipal' => $grantee,
]);
```

Node.js

Weitere Informationen finden Sie in der listGrants-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Grantee = 'arn:aws:iam::111122223333:role/test-engineer';

kmsClient.listGrants({ KeyId, Grantee }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Get-KMS-Cmdlet, um die Details aller AWS KMS Erteilungen für einen KMS-Schlüssel anzuzeigen. GrantList

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
$grantee = 'arn:aws:iam::111122223333:role/test-engineer'
$response = Get-KMSGrantList -KeyId $keyId -GranteePrincipal $grantee
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Aufheben einer Erteilung

Um eine Erteilung für einen KMS-Schlüssel aufzuheben, verwenden Sie die -RetireGrantOperation. Nicht mehr benötigte Erteilungen sollten immer aufgehoben werden.

Wenn Sie eine Erteilung aufheben möchten, geben Sie das Erteilungs-Token oder sowohl die Erteilungs-ID als auch die KMS-Schlüssel-ID an. Für diesen Vorgang muss die KMS-Schlüssel-ID der Amazon-Ressourcenname (ARN) des KMS-Schlüssels sein. Das Erteilungs-Token wird von der -CreateGrantOperation zurückgegeben. Die Erteilungs-ID wird von den ListGrants Operationen CreateGrant und zurückgegeben.

RetireGrant gibt keine Antwort zurück. Um zu überprüfen, ob es wirksam war, verwenden Sie die -ListGrantsOperation.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der retireGrant-Methode in der AWS SDK for Java-API-Referenz.

```
// Retire a grant
//
String grantToken = Place your grant token here;

RetireGrantRequest req = new RetireGrantRequest().withGrantToken(grantToken);
kmsClient.retireGrant(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der RetireGrant encrypt-Methode im AWS SDK for .NET.

```
// Retire a grant
//
String grantToken = "Place your grant token here";

RetireGrantRequest retireGrantRequest = new RetireGrantRequest()
{
    GrantToken = grantToken
};
kmsClient.RetireGrant(retireGrantRequest);
```

Python

Weitere Informationen finden Sie in der Beschreibung der `retire_grant`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Retire a grant

grant_token = Place your grant token here

response = kms_client.retire_grant(
    GrantToken=grant_token
)
```

Ruby

Weitere Informationen finden Sie unter der `retire_grant`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Retire a grant

grant_token = Place your grant token here

response = kmsClient.retire_grant({
  grant_token: grant_token
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der RetireGrantencrypt-Methode im AWS SDK for PHP.

```
// Retire a grant
```

```
//
$grantToken = 'Place your grant token here';

$result = $KmsClient->retireGrant([
    'GrantToken' => $grantToken,
]);
```

Node.js

Weitere Informationen finden Sie in der retireGrant-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Retire a grant
//
const GrantToken = 'Place your grant token here';
kmsClient.retireGrant({ GrantToken }, (err, data) => {
  ...
});
```

PowerShell

Verwenden Sie das Cmdlet Disable-KMSGrant, um eine Erteilung aufzuheben. Verwenden Sie das Cmdlet New-KMSGrant, um das Erteilungs-Token abzurufen. Der `GrantToken`-Parameter verwendet eine Zeichenfolge, sodass Sie keine Ausgabe konvertieren müssen, die vom Cmdlet Read-Host zurückgegeben wird.

```
# Retire a grant

$grantToken = Read-Host -Message Place your grant token here
Disable-KMSGrant -GrantToken $grantToken
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Zurückziehen einer Erteilung

Um eine Erteilung für einen KMS-Schlüssel zu widerrufen, verwenden Sie die -RevokeGrantOperation. Sie können eine Erteilung widerrufen, um ausdrücklich Produktionen abzulehnen, die diese Erteilung unbedingt benötigen.

In Sprachen, für die ein Client-Objekt erforderlich ist, verwenden diese Beispiele das AWS KMS-Client-Objekt, das Sie in Erstellen eines Clients erstellt haben.

Java

Weitere Informationen finden Sie in der Beschreibung der revokeGrant-Methode in der AWS SDK for Java-API-Referenz.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest req = new
 RevokeGrantRequest().withKeyId(keyId).withGrantId(grantId);
kmsClient.revokeGrant(req);
```

C#

Weitere Informationen finden Sie in der Beschreibung der RevokeGrant encrypt-Methode im AWS SDK for .NET.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest revokeGrantRequest = new RevokeGrantRequest()
{
    KeyId = keyId,
    GrantId = grantId
};
kmsClient.RevokeGrant(revokeGrantRequest);
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Python

Weitere Informationen finden Sie in der Beschreibung der `revoke_grant`encrypt-Methode im AWS SDK for Python (Boto3).

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kms_client.revoke_grant(
    KeyId=key_id,
    GrantId=grant_id
)
```

Ruby

Weitere Informationen finden Sie unter der `revoke_grant`-Instance-Methode encrypt im AWS SDK for Ruby.

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kmsClient.revoke_grant({
  key_id: key_id,
  grant_id: grant_id
})
```

PHP

Weitere Informationen finden Sie in der Beschreibung der RevokeGrantencrypt-Methode im AWS SDK for PHP.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
$grantId = "grant1";

$result = $KmsClient->revokeGrant([
    'KeyId' => $keyId,
    'GrantId' => $grantId,
]);
```

Node.js

Weitere Informationen finden Sie in der revokeGrant-Eigenschaft im AWS -SDK für JavaScript in Node.js.

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
const GrantId = 'grant1';
kmsClient.revokeGrant({ GrantId, KeyId }, (err, data) => {
  ...
});
```

PowerShell

Um eine Erteilung zu widerrufen, verwenden Sie das Cmdlet Revoke-KMSGrant.

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
```

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

# Replace the following example grant ID with a valid one
$grantId = 'grant1'

Revoke-KMSGrant -KeyId $keyId -GrantId $grantId
```

Um die AWS KMS PowerShell Cmdlets zu verwenden, installieren Sie das AWS.Tools.KeyManagementService-Modul. Weitere Informationen finden Sie im AWS Tools for Windows PowerShell-Benutzerhandbuch.

Testen Ihrer AWS KMS-API-Aufrufe

Um AWS KMS zu verwenden, müssen Sie über Anmeldeinformationen verfügen, die AWS zur Authentifizierung Ihrer API-Anforderungen verwenden kann. Die Anmeldedaten müssen die Berechtigung zum Zugriff auf KMS-Schlüssel und Aliase enthalten. Die Berechtigungen werden durch Schlüsselrichtlinien, IAM-Richtlinien, Zuschüsse und kontoübergreifende Zugriffskontrollen bestimmt. Sie können nicht nur den Zugriff auf KMS-Schlüssel steuern, sondern auch den Zugriff auf Ihr CloudHSM und auf Ihre benutzerdefinierten Schlüsselspeicher.

Sie können den `DryRun`-API-Parameter angeben, um zu überprüfen, ob Sie über die erforderlichen Berechtigungen verfügen, um AWS KMS-Schlüssel zu verwenden. Sie können `DryRun` auch verwenden, um zu überprüfen, ob die Anforderungsparameter in einem AWS KMS-API-Aufruf korrekt angegeben sind.

Themen

- Was ist der - DryRun Parameter?
- Angeben DryRun mit der API

Was ist der - DryRun Parameter?

`DryRun` ist ein optionaler API-Parameter, den Sie angeben, um zu überprüfen, ob AWS KMS-API-Aufrufe erfolgreich sind. Verwenden Sie `DryRun`, um Ihren API-Aufruf zu testen, bevor Sie AWS KMS tatsächlich aufrufen. Sie können die folgenden Punkte überprüfen.

- Dass Sie über die erforderlichen Berechtigungen verfügen, um AWS KMS-Schlüssel zu verwenden.

- Dass Sie die Parameter im Aufruf korrekt angegeben haben.

AWS KMS unterstützt die Verwendung des `DryRun`-Parameters in bestimmten API-Aktionen:

- CreateGrant
- Decrypt
- Encrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign
- Verify
- VerifyMac

Die Verwendung des `DryRun`-Parameters ist kostenpflichtig und wird als Standard-API-Anfrage in Rechnung gestellt. Weitere Informationen zu AWS KMS-Preisen erhalten Sie unter AWS Key Management Service – Preise.

Alle API-Anfragen, die den `DryRun`-Parameter verwenden, beziehen sich auf das Anforderungskontingent der API und können zu einer Drosselungsausnahme führen, wenn Sie ein API-Anforderungskontingent überschreiten. Beispielsweise wird der Aufruf von Decrypt mit `DryRun` oder ohne `DryRun` demselben Kontingent für kryptografische Operationen angerechnet. Weitere Informationen hierzu finden Sie unter Drosselung AWS KMS von Anfragen.

Jeder Aufruf an einen AWS KMS-API-Vorgang wird als Ereignis erfasst und in einem AWS CloudTrail-Protokoll aufgezeichnet. Die Ausgabe aller Operationen, die den `DryRun` Parameter angeben, wird in Ihrem CloudTrail Protokoll angezeigt. Weitere Informationen finden Sie unter AWS KMS API-Aufrufe protokollieren mit AWS CloudTrail.

Angeben DryRun mit der API

Um `DryRun` zu verwenden, geben Sie den Parameter `–dry-run` in AWS CLI-Befehlen und AWS KMS-API-Aufrufen an, die den Parameter unterstützen. Wenn Sie dies tun, prüft AWS KMS, ob Ihr Aufruf erfolgreich sein wird. AWS KMS-Aufrufe, die `DryRun` verwenden, schlagen immer fehl und geben eine Meldung mit Informationen über den Grund für das Scheitern des Aufrufs zurück. Die Nachricht kann die folgenden Ausnahmen enthalten:

- `DryRunOperationException` – Die Anfrage wäre erfolgreich, wenn `DryRun` nicht angegeben wäre.
- `ValidationException` – Die Anfrage schlug fehl, weil ein falscher API-Parameter angegeben wurde.
- `AccessDeniedException` – Sie sind nicht berechtigt, die angegebene API-Aktion auf der KMS-Ressource auszuführen.

Der folgende Befehl verwendet beispielsweise die -CreateGrantOperation und erstellt eine Erteilung, die es Benutzern, die die `keyUserRole` Rolle annehmen dürfen, ermöglicht, die Decrypt-Operation für einen angegebenen symmetrischen KMS-Schlüssel aufzurufen. Der `DryRun`-Parameter ist angegeben.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

AWS KMS eventuelle Datenkonsistenz

Die AWS KMS-API folgt aufgrund der verteilten Natur des Systems einem Modell der eventuellen Konsistenz. Dies hat zur Folge, dass Änderungen an AWS KMS-Ressourcen möglicherweise nicht sofort für nachfolgende Befehle, die Sie ausführen, sichtbar sind.

Wenn Sie AWS KMS-API-Aufrufe ausführen, kann es zu einer kurzen Verzögerung kommen, bis die Änderung in AWS KMS verfügbar ist. In der Regel dauert es weniger als ein paar Sekunden, bis sich die Änderung im gesamten System verbreitet, in einigen Fällen kann es jedoch mehrere Minuten dauern. Während dieser Zeit können unerwartete Fehler auftreten, wie z. B. eine `NotFoundException` oder eine `InvalidStateException`. Zum Beispiel könnte AWS KMS eine

`NotFoundException` zurückgeben, wenn Sie `GetParametersForImport` unmittelbar nach dem Aufruf von `CreateKey` aufrufen.

Wir empfehlen Ihnen, auf Ihren AWS KMS-Clients eine Wiederholungsstrategie zu konfigurieren, sodass Vorgänge nach einer kurzen Wartezeit automatisch wiederholt werden. Weitere Informationen finden Sie unter Wiederholungsverhalten im Referenzhandbuch zu AWS-SDKs und Tools.

Für API-Aufrufe im Zusammenhang mit Zuschüssen können Sie ein Grant-Token verwenden, um mögliche Verzögerungen zu vermeiden, und die in einem Grant enthaltenen Berechtigungen sofort nutzen. Weitere Informationen finden Sie unter Letztendliche Konsistenz (für Erteilungen).

Referenzen

Die folgenden Referenzen umfassen hilfreiche Informationen zur Verwendung und Verwaltung von KMS-Schlüsseln.

- Schlüsseltypreferenz. Führt den Typ des KMS-Schlüssels auf, der die jeweilige AWS KMS-API-Operation unterstützt.

  Suche: Kann ich einen RSA-Signatur-KMS-Schlüssel aktivieren und deaktivieren?
- Schlüsselstatus-Tabelle. Zeigt, wie der Schlüsselstatus eines KMS-Schlüssels dessen Verwendung in AWS KMS-API-Operationen beeinflusst.

  Suche: Kann ich den Alias eines KMS-Schlüssels ändern, dessen Löschung aussteht?
- Referenz zu AWS KMS-API-Berechtigungen. Stellt Informationen zu den erforderlichen Berechtigungen für die jeweiligen AWS KMS-API-Operation bereit.

  Suche: Kann ich GetKeyPolicy auf einem Schlüssel in einem anderen AWS Konto ausführen? Kann ich die Berechtigung `kms:Decrypt` in einer IAM-Richtlinie erlauben?
  - ViaService Referenz zu . Führt die AWS-Services auf, die den Bedingungsschlüssel `kms:ViaService` unterstützen.

    Suche: Kann ich den `kms:ViaService` Bedingungsschlüssel verwenden, um eine Berechtigung nur zu erteilen, wenn sie von Amazon stammt ElastiCache? Und wie ist das bei Amazon Neptune?
- AWS KMS – Preise. Führt die Preise für KMS-Schlüssel auf und erklärt sie.

  Suche: Wie viel kostet die Verwendung meiner asymmetrischen Schlüssel?
- Anforderungskontingente in AWS KMS. Führt die sekundengenauen Kontingente für AWS KMS-API-Anforderungen in jedem Konto und jeder Region auf.

  Suche: Wie viele Decrypt-Anforderungen kann ich pro Sekunde ausführen? Wie viele Decrypt-Anforderungen kann ich auf KMS-Schlüsseln in meinem benutzerdefinierten Schlüsselspeicher ausführen?
- Ressourcenkontingente in AWS KMS. Führt die Kontingente für AWS KMS-Ressourcen auf.

  Suche: Wie viele KMS-Schlüssel kann ich in den einzelnen Regionen meines Kontos haben? Wie viele Aliasse kann ich für jeden KMS-Schlüssel haben?

- AWS-Services integriert mit AWS KMS. Führt die AWS-Services auf, die KMS-Schlüssel verwenden, um die Ressourcen zu schützen, die sie erstellen, speichern und verwalten.

  Suche: Verwendet Amazon Connect KMS-Schlüssel, um meine Connect-Ressourcen zu schützen?

Dokumentverlauf

In diesem Thema werden wichtige Aktualisierungen im AWS Key Management Service - Entwicklerhandbuch beschrieben.

Themen

Neueste Aktualisierungen

Die folgende Tabelle beschreibt signifikante Änderungen an dieser Dokumentation seit Januar 2018. Neben den hier aufgelisteten größeren Änderungen aktualisieren wir die Dokumentation regelmäßig überarbeitet, um Beschreibungen und Beispiele zu verbessern und Ihre Rückmeldungen zu berücksichtigen. Wenn Sie über wichtige Änderungen benachrichtigt werden möchten, abonnieren Sie den RSS-Feed.

Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in dieser Tabelle anzuzeigen.

Änderung	Beschreibung	Datum
Aktualisierungen der Schlüsselrotation	Unterstützung für benutzerdefinierte Rotationsperioden für automatische Schlüsselrotationen, On-Demand-Schlüsselrotationen und Einblick in Ihre wichtigsten Materialrotationen hinzugefügt.	12. April 2024
Aktualisierungen der verwalteten Richtlinie	Es wurden neue Berechtigungen hinzugefügt`AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` , die es AWS KMS ermöglichen, Änderunge	10. November 2023

	n in der VPC, die Ihren AWS CloudHSM Cluster enthält, zu überwachen, sodass bei Ausfällen klare Fehlermel dungen ausgegeben werden AWS KMS können.	
Feature-Update	Unterstützung für den `DryRun`-API-Parameter hinzugefügt.	5. Juli 2023
Feature-Update	Unterstützung für den Import von Schlüsselmaterial für alle Arten von AWS KMS Schlüsseln hinzugefügt, mit Ausnahme von benutzerd efinierten Schlüsselspeichern.	5. Juni 2023
Feature-Update	Aktualisierungen der AWS KMS APIs für Nitro Enclaves	10. März 2023
Feature-Update	Der `RSAES_PKCS1_V1_5` Wrapping-Algorithmus ist veraltet. AWS KMS wird gemäß den Richtlinien des National Institute of Standards and Technology (NIST) zur Verwaltung kryptografischer Schlüssel bis zum 1. Oktober 2023 jegliche Unterstüt zung einstellen. `RSAES_PKC S1_V1_5` Wir empfehlen , sofort mit der Verwendun g eines anderen Wrapping-Algorithmus zu beginnen.	28. Februar 2023

Feature-Update	Zusätzliche Unterstützung für externe Schlüsselspeicher, eine Funktion, mit der Sie Ihre AWS Ressourcen schützen können, indem Sie kryptografische Schlüssel außerhalb von verwenden. AWS	29. November 2022
Kontingentänderung	Das AWS KMS keys Ressourcenkontingent wurde für jedes Konto und jede Region auf 100.000 KMS-Schlüssel erhöht.	8. Juli 2022
Funktionsupdate	Unterstützung für HMAC-KMS-Schlüssel wurde in weiteren Bereichen hinzugefügt AWS-Regionen	8. Juli 2022
Neues Thema	Das AWS Key Management Service Thema Resilienz wurde dem Kapitel Sicherheit des AWS KMS Entwicklerhandbuchs hinzugefügt.	14. Juni 2022
Neues Feature	Unterstützung für AWS KMS Schlüssel und API-Operationen, die HMAC-Codes generieren und verifizieren, wurde hinzugefügt.	19. April 2022
Änderung der Dokumentation	Ersetzen des Begriffes Kundenhauptschlüssel (Customer Master Key, CMK) mit AWS KMS key und KMS-Schlüssel.	30. August 2021

Neue Funktion	Zusätzliche Unterstützung für multiregionale Schlüssel, ein Satz interoperabler KMS-Schlüssel in verschiedenen Regionen, die die gleiche Schlüssel-ID und das gleiche Schlüsselmaterial haben. Sie können multiregionale Schlüssel verwenden, um Daten in einer Region zu verschlüsseln und in einer beliebigen anderen Region zu entschlüsseln.	8. Juni 2021
Neue Funktion	Zusätzliche Unterstützung für attributbasierte Zugriffss teuerung (ABAC). Sie können Tags und Aliase verwenden , um den Zugriff auf Ihre zu kontrollieren. AWS KMS keys	17. Dezember 2020
Neue Funktion	Zusätzliche Unterstützung für VPC-Endpunktrichtlinien.	9. Juli 2020
Neuer Inhalt	Erläutert die Sicherhei tseigenschaften von. AWS KMS	18. Juni 2020
Neue Funktion	Unterstützung für asymmetri sche AWS KMS keys und asymmetrische Datenschl üssel wurde hinzugefügt.	25. November 2019

Aktualisiertes Feature	Sie können die Schlüssel richtlinie von Von AWS verwaltete Schlüssel in der AWS KMS Konsole einsehen. Diese Funktion war früher auf kundenverwaltete Schlüssel beschränkt.	15. November 2019
Neue Funktion	Erläutert, wie Hybrid-Post-Quantum-Schlüsselaustauschalgorithmen in TLS für Ihre Aufrufe an AWS KMS verwendet werden.	4. November 2019
Kontingentänderung	Für einige APIs, die KMS-Schlüssel verwalten, wurden die Ressourcenkontingente erhöht.	18. September 2019
Kontingentänderung	Die Ressourcenkontingente für KMS-Schlüssel, Aliasse und Erteilungen pro KMS-Schlüssel wurden geändert.	27. März 2019
Kontingentänderung	Das gemeinsame pro Sekunde Anforderungskontingent für kryptografische Operationen, die AWS KMS keys in einem benutzerdefinierten Schlüsselspeicher verwenden, wurde geändert.	7. März 2019

Neue Funktion	Erläutert, wie AWS KMS benutzerdefinierte Schlüsselspeicher erstellt und verwaltet werden. Jeder Schlüsselspeicher wird von einem AWS CloudHSM Cluster unterstützt, den Sie besitzen und kontrollieren.	26. November 2018
New console	Erläutert, wie die neue AWS KMS Konsole verwendet wird, die unabhängig von der IAM-Konsole ist. Die ursprüngliche Version der Konsole wird zusammen mit der Anleitung zu ihrer Verwendung noch eine kurze Zeit lang zur Verfügung stehen, um Ihnen Zeit zu geben, sich mit der neuen Konsole vertraut zu machen.	7. November 2018
Kontingentänderung	Das Kontingent für gemeinsame Anfragen zur Verwendung von AWS KMS keys wurde geändert.	21. August 2018
Neuer Inhalt	Erläutert, wie AWS KMS Schlüssel AWS Secrets Manager verwendet werden, um den geheimen Wert in einem Geheimnis zu verschlüsseln.	13. Juli 2018

Neuer Inhalt	Erläutert, wie DynamoDB AWS KMSAWS KMS keys verwendet, um die serversei tige Verschlüsselungs-Option zu unterstützen.	23. Mai 2018
Neue Funktion	Erläutert, wie Sie einen privaten Endpunkt in Ihrer VPC verwenden, um eine direkte Verbindung herzustel len AWS KMS, anstatt eine Verbindung über das Internet herzustellen.	22. Januar 2018

Frühere Aktualisierungen

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Key Management Service Entwicklerhandbuch vor 2018 beschrieben.

Möglicherweise müssen Sie horizontal oder vertikal scrollen, um alle Daten in dieser Tabelle anzuzeigen.

Änderung	Beschreibung	Datum
Neuer Inhalt	Dokumentation zu Tagging von Schlüsseln hinzugefügt.	15. Februar 2017
Neuer Inhalt	Dokumentation zu Überwachu ng von AWS KMS keys und Überwachung mit Amazon CloudWatch hinzugefügt.	31. August 2016
Neuer Inhalt	Dokumentation zu Importiertes Schlüsselmaterial hinzugefügt.	11. August 2016
Neuer Inhalt	Die folgende Dokumentation wurde hinzugefügt: IAM-Richt	5. Juli 2016

Änderung	Beschreibung	Datum
	linien, Berechtigungsreferenz und Bedingungsschlüssel.	
Aktualisierung	Aktualisierung von Teilen der Dokumentation im Kapitel Authentifizierung und Zugriffskontrolle.	5. Juli 2016
Aktualisierung	Die Seite Kontingente wurde aktualisiert, um die neuen Standardkontingente zu berücksichtigen.	31. Mai 2016
Aktualisierung	Die Seite Kontingente wurde aktualisiert, um die neuen Standardkontingente zu berücksichtigen. Außerdem wurde die Erteilungs-Token-Dokumentation aktualisiert, um sie zu verdeutlichen und genauer zu gestalten.	11. April 2016
Neuer Inhalt	Dokumentation zu Mehreren IAM-Prinzipalen Zugriff auf einen KMS-Schlüssel gewähren und Verwenden der IP-Adressbedingung hinzugefügt.	17. Februar 2016
Aktualisierung	Aktualisierung der Seiten Wichtige Richtlinien in AWS KMS und Ändern einer Schlüsselrichtlinie zur Verbesserung der Klarheit und Genauigkeit.	17. Februar 2016

Änderung	Beschreibung	Datum
Aktualisierung	Aktualisierung der Schlüssel verwalten-Themenseiten zur Verbesserung der Klarheit.	5. Januar 2016
Neuer Inhalt	Dokumentation zu Wie AWS CloudTrailAWS KMS verwendet hinzugefügt.	18. November 2015
Neuer Inhalt	Ergänzung von Anweisungen für Ändern einer Schlüsselrichtlinie.	18. November 2015
Aktualisierung	Dokumentation über Wie Amazon Relational Database Service (Amazon RDS) AWS KMS nutzt aktualisiert.	18. November 2015
Neuer Inhalt	Dokumentation zu Wie WorkSpaces verwendet AWS KMS hinzugefügt.	6. November 2015
Aktualisierung	Aktualisierung der Seite Wichtige Richtlinien in AWS KMS zur Verbesserung der Klarheit.	22. Oktober 2015
Neuer Inhalt	Ergänzung von Dokumentation zu Löschen von AWS KMS keys, einschließlich Begleitdokumentation zu Erstellen eines Alarms und Feststellen der früheren Nutzung eines KMS-Schlüssels.	15. Oktober 2015

Änderung	Beschreibung	Datum
Neuer Inhalt	Dokumentation zu Bestimmen des Zugriffs auf AWS KMS keys hinzugefügt.	15. Oktober 2015
Neuer Inhalt	Dokumentation zu Wichtige Zustände von AWS KMS Schlüsseln hinzugefügt.	15. Oktober 2015
Neuer Inhalt	Dokumentation zu Wie Amazon Simple Email Service (Amazon SES) AWS KMS nutzt. hinzugefügt.	1. Oktober 2015
Aktualisierung	Die Seite Kontingente wurde aktualisiert, um die neuen Anforderungskontingente zu erläutern.	31. August 2015
Neuer Inhalt	Es wurden Informationen zu den Nutzungsgebühren hinzugefügt AWS KMS. Siehe AWS KMS -Preise	14. August 2015
Neuer Inhalt	Anforderungskontingente wurden dem hinzugefügt AWS KMS Kontingente.	11. Juni 2015
Neuer Inhalt	Neues Java-Code-Beispiel zur Demonstration der Verwendung der Produktion `UpdateAlias` hinzugefügt. Siehe Aktualisieren eines Alias.	1. Juni 2015

Änderung	Beschreibung	Datum
Aktualisierung	AWS Key Management Service -Regionentabelle zur Allgemeine AWS-Referenz verschoben.	29. Mai 2015
Neuer Inhalt	Dokumentation zu Wie Amazon EMR AWS KMS nutzt hinzugefügt.	28. Januar 2015
Neuer Inhalt	Dokumentation zu So WorkMail verwendet Amazon AWS KMS hinzugefügt.	28. Januar 2015
Neuer Inhalt	Dokumentation zu Wie Amazon Relational Database Service (Amazon RDS) AWS KMS nutzt hinzugefügt.	6. Januar 2015
Neuer Inhalt	Dokumentation zu Wie Amazon Elastic Transcoder AWS KMS nutzt hinzugefügt.	24. November 2014
Neues Handbuch	Einführung des AWS Key Management Service - Entwicklerhandbuchs.	12. November 2014

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