aws

Panduan Developer

AWS Key Management Service

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Key Management Service: Panduan Developer

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

AWS Key Management Service

AWS Key Management Service(AWS KMS) adalah layanan terkelola yang memudahkan Anda membuat dan mengontrol kunci kriptografi yang digunakan untuk melindungi data Anda. AWS KMSmenggunakan modul keamanan perangkat keras (HSM) untuk melindungi dan memvalidasi Anda AWS KMS keys di bawah Program Validasi Modul Kriptografi FIPS 140-2. Wilayah China (Beijing) dan China (Ningxia) tidak mendukung Program Validasi Modul Kriptografi FIPS 140-2. AWS KMSmenggunakan HSM bersertifikat OSCCA untuk melindungi kunci KMS di Wilayah China.

AWS KMSterintegrasi dengan sebagian besar AWSlayanan lain yang mengenkripsi data Anda. AWS KMSjuga terintegrasi dengan AWS CloudTrailuntuk mencatat penggunaan kunci KMS Anda untuk kebutuhan audit, peraturan, dan kepatuhan.

Anda dapat menggunakan AWS KMS API untuk membuat dan mengelola kunci KMS dan fitur khusus, seperti toko kunci khusus, dan menggunakan kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Referensi AWS Key Management Service API.

Anda dapat membuat dan mengelolaAWS KMS keys:

- Buat, edit, dan lihat tombol KMS simetris dan asimetris, termasuk tombol HMAC.
- Kontrol akses ke kunci KMS Anda dengan menggunakan kebijakan utama, kebijakanIAM, dan hibah. AWS KMSmendukung kontrol akses berbasis atribut (ABAC). Anda juga dapat menyempurnakan kebijakan dengan menggunakan kunci kondisi.
- Buat, hapus, daftar, dan perbarui alias, nama ramah untuk kunci KMS Anda. Anda juga dapat menggunakan alias untuk mengontrol akses ke kunci KMS Anda.
- Tandai kunci KMS Anda untuk identifikasi, otomatisasi, dan pelacakan biaya. Anda juga dapat menggunakan tag untuk mengontrol akses ke kunci KMS Anda.
- Aktifkan dan nonaktifkan tombol KMS.
- Aktifkan dan nonaktifkan rotasi otomatis bahan kriptografi dalam kunci KMS.
- Hapus kunci KMS untuk menyelesaikan siklus hidup kunci.

Anda dapat menggunakan kunci KMS Anda dalam operasi kriptografi. Sebagai contoh, lihat Memprogram API AWS KMS.

- Mengenkripsi, mendekripsi, dan mengenkripsi ulang data dengan kunci KMS simetris atau asimetris.

- Tanda tangani dan verifikasi pesan dengan tombol KMS asimetris.
- Hasilkan kunci data simetris yang dapat diekspor dan pasangan kunci data asimetris.
- Hasilkan dan verifikasi kode HMAC.
- Hasilkan angka acak yang cocok untuk aplikasi kriptografi.

Anda dapat menggunakan fitur AWS KMS lanjutan.

- Buat kunci Multi-wilayah, yang bertindak seperti salinan kunci KMS yang sama di berbeda. Wilayah AWS
- Impor materi kriptografi ke dalam kunci KMS.
- Buat kunci KMS di toko AWS CloudHSM kunci yang didukung oleh AWS CloudHSM cluster Anda.
- Buat kunci KMS di toko kunci eksternal yang didukung oleh kunci kriptografi Anda di luar. AWS
- Connect langsung ke AWS KMS melalui endpoint pribadi di VPC Anda.
- Gunakan TLS pasca-kuantum hibrida untuk menyediakan enkripsi berwawasan ke depan dalam perjalanan untuk data yang Anda kirim. AWS KMS

Dengan menggunakan AWS KMS, Anda mendapatkan lebih banyak kontrol atas akses ke data yang dienkripsi. Anda dapat menggunakan manajemen kunci dan fitur kriptografi langsung di aplikasi Anda atau melalui AWS layanan yang terintegrasi dengannyaAWS KMS. Apakah Anda menulis aplikasi untuk AWS atau menggunakan AWS layanan, AWS KMS memungkinkan Anda untuk mempertahankan kontrol atas siapa yang dapat menggunakan Anda AWS KMS keys dan mendapatkan akses ke data terenkripsi Anda.

AWS KMSterintegrasi denganAWS CloudTrail, layanan yang mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Dengan menggunakan CloudTrail Anda dapat memantau dan menyelidiki bagaimana dan kapan kunci KMS Anda telah digunakan dan siapa yang menggunakannya.

AWS KMS dalam Wilayah AWS

Wilayah AWS yang mendukung AWS KMS tercantum dalam Titik akhir dan Kuota AWS Key Management Service. Jika fitur AWS KMS tidak didukung dalam Wilayah AWS yang mendukung AWS KMS, perbedaan regional dijelaskan dalam topik tentang fitur.

Harga AWS KMS

Seperti halnya AWS produk lain, penggunaan AWS KMS tidak memerlukan kontrak atau pembelian minimum. Untuk informasi lebih lanjut tentang harga AWS KMS, lihat Harga AWS Key Management Service.

Perjanjian tingkat layanan

AWS Key Management Service didukung oleh perjanjian tingkat layanan yang menentukan kebijakan ketersediaan layanan kami.

Pelajari selengkapnya

- Untuk mempelajari istilah dan konsep yang digunakan dalam AWS KMS, lihat Konsep AWS KMS.
- Untuk informasi tentang AWS KMS API, lihat Referensi API AWS Key Management Service. Untuk contoh dalam bahasa pemrograman lainnya, lihat Memprogram API AWS KMS.
- Untuk mempelajari cara menggunakan AWS CloudFormation templat untuk membuat dan mengelola kunci dan alias, lihat Menciptakan AWS KMS sumber daya dengan AWS CloudFormation dan referensi jenis AWS Key Management Service sumber daya di Panduan AWS CloudFormation Pengguna.
- Untuk informasi teknis terperinci tentang cara AWS KMS menggunakan kriptografi dan mengamankan kunci KMS, lihat AWS Key Management Service Detail Kriptografi. Dokumentasi Detail Kriptografis tidak menjelaskan cara kerja AWS KMS di Wilayah China (Beijing) dan China (Ningxia).
- Untuk daftar AWS KMS titik akhir, termasuk titik akhir FIPS, di masing-masing titikWilayah AWS, lihat Titik akhir layanan dalam topik. AWS Key Management Service Referensi Umum AWS
- Untuk bantuan terkait pertanyaan tentang AWS KMS, lihat Forum Diskusi AWS Key Management Service.

AWS KMSdi AWS SDK

- 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

Konsep AWS KMS

Pelajari istilah dan konsep dasar yang digunakan dalam AWS Key Management Service (AWS KMS) dan bagaimana mereka bekerja sama untuk membantu melindungi data Anda.

Topik

- AWS KMS keys
- Kunci dan AWS kunci pelanggan
- Kunci KMS enkripsi simetris
- Tombol Asymmetric KMS
- Kunci HMAC KMS
- Kunci data
- Pasangan kunci data
- Alias
- Penyimpanan kunci kustom
- Operasi kriptografi
- Pengidentifikasi kunci () KeyId
- Material kunci
- Asal material kunci
- Spesifikasi kunci
- Penggunaan kunci
- Enkripsi amplop
- Konteks enkripsi
- Kebijakan kunci
- Pemberian Izin
- Mengaudit penggunaan kunci KMS
- Infrastruktur manajemen kunci

AWS KMS keys

AWS KMS keys(Kunci KMS) adalah sumber daya utama diAWS KMS. Anda dapat menggunakan kunci KMS untuk mengenkripsi, mendekripsi, dan mengenkripsi ulang data. Ini juga dapat menghasilkan kunci data yang dapat Anda gunakan di luar AWS KMS. Biasanya, Anda akan menggunakan kunci KMS enkripsi simetris, tetapi Anda dapat membuat dan menggunakan kunci KMS asimetris untuk enkripsi atau penandatanganan, dan membuat dan menggunakan kunci KMS HMAC untuk menghasilkan dan memverifikasi tag HMAC.

> **Note**
>
> AWS KMSmengganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS. Konsepnya tidak berubah. Untuk mencegah perubahan yang melanggar, AWS KMS adalah menjaga beberapa variasi dari istilah ini.

An AWS KMS keyadalah representasi logis dari kunci kriptografi. Kunci KMS berisi metadata, seperti ID kunci, spesifikasi kunci, penggunaan kunci, tanggal pembuatan, deskripsi, dan status kunci. Yang paling penting, ini berisi referensi ke bahan utama yang digunakan ketika Anda melakukan operasi kriptografi dengan kunci KMS.

Anda dapat membuat kunci KMS dengan materi kunci kriptografi yang dihasilkan dalam modul keamanan perangkat keras yang divalidasi AWS KMS FIPS. Bahan kunci untuk kunci KMS simetris dan kunci pribadi kunci KMS asimetris tidak pernah dibiarkan tidak terenkripsi. AWS KMS Untuk menggunakan atau mengelola kunci KMS Anda, Anda harus menggunakanAWS KMS. Untuk informasi tentang membuat dan mengelola kunci KMS, lihatMengelola kunci. Untuk informasi tentang penggunaan kunci KMS, lihat Referensi AWS Key Management Service API.

Secara default, AWS KMS membuat materi kunci untuk kunci KMS. Anda tidak dapat mengekstrak, mengekspor, melihat, atau mengelola material kunci ini. Satu-satunya pengecualian adalah kunci publik dari asymmetric key pair, yang dapat Anda ekspor untuk digunakan di luar. AWS Selain itu, Anda tidak dapat menghapus materi kunci ini; Anda harus menghapus kunci KMS. Namun, Anda dapat mengimpor materi kunci Anda sendiri ke dalam kunci KMS, atau menggunakan penyimpanan kunci khusus untuk membuat kunci KMS yang menggunakan materi kunci di AWS CloudHSM klaster Anda, atau materi kunci di pengelola kunci eksternal yang Anda miliki dan kelola di luar. AWS

AWS KMSjuga mendukung kunci Multi-wilayah, yang memungkinkan Anda mengenkripsi data dalam satu Wilayah AWS dan mendekripsi dalam yang berbeda. Wilayah AWS

Untuk informasi tentang membuat dan mengelola kunci KMS, lihatMengelola kunci. Untuk informasi tentang penggunaan kunci KMS, lihat Referensi AWS Key Management Service API.

Kunci dan AWS kunci pelanggan

Kunci KMS yang Anda buat adalah kunci yang dikelola pelanggan. Layanan AWSyang menggunakan kunci KMS untuk mengenkripsi sumber daya layanan Anda sering membuat kunci untuk Anda. Kunci KMS yang Layanan AWS dibuat di AWS akun Anda adalah Kunci yang dikelola AWS. Kunci KMS yang Layanan AWS dibuat di akun layanan adalah Kunci milik AWS.

Jenis kunci KMS	Dapat melihat metadata kunci KMS	Dapat mengelola kunci KMS	Digunakan hanya untuk Akun AWS saya	Rotasi otomatis	Penetapan Harga
Kunci yang dikelola pelanggan	Ya	Ya	Ya	Opsional. Setiap tahun (sekitar 365 hari)	Biaya bulanan (pro-rated per jam) Biaya per penggunaan
Kunci yang dikelola AWS	Ya	Tidak	Ya	Wajib. Setiap tahun (sekitar 365 hari)	Tidak ada biaya bulanan Biaya per penggunaan (beberapa Layanan AWS membayar biaya ini untuk Anda)
Kunci milik AWS	Tidak	Tidak	Tidak	Bervariasi	Tidak ada biaya

AWSlayanan yang terintegrasi dengan AWS KMS berbeda dalam dukungan mereka untuk kunci KMS. Beberapa AWS layanan mengenkripsi data Anda secara default dengan Kunci milik AWS atau fileKunci yang dikelola AWS. Beberapa AWS layanan mendukung kunci yang dikelola pelanggan. AWSLayanan lain mendukung semua jenis kunci KMS untuk memungkinkan Anda kemudahanKunci milik AWS, visibilitasKunci yang dikelola AWS, atau kontrol kunci yang dikelola pelanggan. Untuk informasi mendetail tentang opsi enkripsi yang ditawarkan layanan AWS, lihat topik Enkripsi Saat Istirahat dalam panduan pengguna atau panduan developer untuk layanan.

Kunci yang dikelola pelanggan

Kunci KMS yang Anda buat adalah kunci yang dikelola pelanggan. Kunci yang dikelola pelanggan adalah kunci KMS Akun AWS yang Anda buat, miliki, dan kelola. Anda memiliki kontrol penuh atas kunci KMS ini, termasuk membuat dan memelihara kebijakan utama mereka, kebijakan IAM, dan hibah, mengaktifkan danmenonaktifkannya, memutar materi kriptografi mereka, menambahkan tag, membuat alias yang merujuk ke kunci KMS, dan menjadwalkan kunci KMS untuk dihapus.

Kunci terkelola pelanggan muncul di halaman kunci terkelola Pelanggan AWS Management Console untukAWS KMS. Untuk mengidentifikasi kunci yang dikelola pelanggan secara definitif, gunakan operasi. DescribeKey Untuk kunci yang dikelola pelanggan, nilai `KeyManager` bidang `DescribeKey` respons adalah`CUSTOMER`.

Anda dapat menggunakan kunci yang dikelola pelanggan Anda dalam operasi kriptografi dan penggunaan audit di AWS CloudTrail log. Selain itu, banyak AWSlayanan yang terintegrasi dengan AWS KMS memungkinkan Anda menentukan kunci yang dikelola pelanggan untuk melindungi data yang disimpan dan dikelola untuk Anda.

Kunci yang dikelola pelanggan dikenakan biaya bulanan dan biaya untuk penggunaan melebihi tingkat gratis. Biaya tersebut dihitung terhadap kuota AWS KMS untuk akun Anda. Untuk detailnya, lihat Harga AWS Key Management Service dan Kuota.

Kunci yang dikelola AWS

Kunci yang dikelola AWSadalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh AWSlayanan yang terintegrasi dengannya AWS KMS.

Beberapa AWS layanan memungkinkan Anda memilih Kunci yang dikelola AWS atau kunci yang dikelola pelanggan untuk melindungi sumber daya Anda dalam layanan itu. Secara umum, kecuali Anda diminta untuk mengontrol kunci enkripsi yang melindungi sumber daya Anda, an Kunci yang dikelola AWS adalah pilihan yang baik. Anda tidak perlu membuat atau mempertahankan kunci atau kebijakan utamanya, dan tidak pernah ada biaya bulanan untuk sebuahKunci yang dikelola AWS.

Anda memiliki izin untuk melihat akun Anda, melihat kebijakan utama mereka, dan mengaudit penggunaannya di AWS CloudTrail log. Kunci yang dikelola AWS Namun, Anda tidak dapat mengubah properti apa punKunci yang dikelola AWS, memutarnya, mengubah kebijakan utamanya, atau menjadwalkannya untuk dihapus. Dan, Anda tidak dapat menggunakan Kunci yang dikelola AWS dalam operasi kriptografi secara langsung; layanan yang membuatnya menggunakannya atas nama Anda.

Kunci yang dikelola AWSmuncul di Kunci yang dikelola AWShalaman AWS Management Console forAWS KMS. Anda juga dapat mengidentifikasi Kunci yang dikelola AWS dengan alias mereka, yang memiliki formataws/*service-name*, sepertiaws/redshift. Untuk mengidentifikasi secara definitifKunci yang dikelola AWS, gunakan operasi. DescribeKey SebabKunci yang dikelola AWS, nilai KeyManager bidang DescribeKey responsnya adalahAWS.

Semua Kunci yang dikelola AWS secara otomatis diputar setiap tahun. Anda tidak dapat mengubah jadwal rotasi ini.

> **Note**
>
> Pada Mei 2022, AWS KMS mengubah jadwal rotasi Kunci yang dikelola AWS dari setiap tiga tahun (sekitar 1.095 hari) menjadi setiap tahun (sekitar 365 hari).
> Kunci yang dikelola AWS baru secara otomatis dirotasi satu tahun setelah dibuat, dan kira-kira setiap tahun setelahnya.
> Kunci yang dikelola AWS yang ada secara otomatis dirotasi satu tahun setelah rotasi terbarunya, dan setiap tahun setelahnya.

Tidak ada biaya bulanan untukKunci yang dikelola AWS. Penggunaan CMK tersebut dapat dikenakan biaya jika melebihi dari tingkat gratis, tetapi beberapa layanan AWS menutup biaya ini untuk Anda. Untuk detail selengkapnya, lihat topik Enkripsi Tidak Aktif di panduan pengguna atau panduan developer untuk layanan tersebut. Untuk detail selengkapnya, lihat Harga AWS Key Management Service.

Kunci yang dikelola AWSjangan dihitung terhadap kuota sumber daya pada jumlah kunci KMS di setiap Wilayah akun Anda. Tetapi ketika digunakan atas nama prinsipal di akun Anda, kunci KMS dihitung terhadap kuota permintaan. Untuk detailnya, lihat Kuota.

Kunci milik AWS

Kunci milik AWSadalah kumpulan kunci KMS yang dimiliki dan dikelola AWS layanan untuk digunakan dalam beberapa. Akun AWS Meskipun tidak Kunci milik AWS ada dalam AndaAkun AWS, AWS layanan dapat menggunakan Kunci milik AWS untuk melindungi sumber daya di akun Anda.

Beberapa AWS layanan memungkinkan Anda memilih Kunci milik AWS atau kunci yang dikelola pelanggan. Secara umum, kecuali Anda diminta untuk mengaudit atau mengontrol kunci enkripsi yang melindungi sumber daya Anda, a Kunci milik AWS adalah pilihan yang baik. Kunci milik AWSbenar-benar gratis (tidak ada biaya bulanan atau biaya penggunaan), mereka tidak dihitung terhadap AWS KMSkuota untuk akun Anda, dan mereka mudah digunakan. Anda tidak perlu membuat atau mempertahankan kunci atau kebijakan utamanya.

Rotasi Kunci milik AWS bervariasi antar layanan. Untuk informasi tentang rotasi tertentuKunci milik AWS, lihat topik Enkripsi saat Istirahat di panduan pengguna atau panduan pengembang untuk layanan.

Kunci KMS enkripsi simetris

Saat Anda membuatAWS KMS key, secara default, Anda mendapatkan kunci KMS untuk enkripsi simetris. Ini adalah jenis kunci KMS dasar dan paling umum digunakan.

DalamAWS KMS, kunci KMS enkripsi simetris mewakili kunci enkripsi AES-GCM 256-bit, kecuali di Wilayah China, di mana ia mewakili kunci enkripsi SM4 128-bit. Bahan kunci simetris tidak pernah dibiarkan tidak AWS KMS terenkripsi. Untuk menggunakan kunci KMS enkripsi simetris, Anda harus menelepon. AWS KMS Kunci enkripsi simetris digunakan dalam enkripsi simetris, di mana kunci yang sama digunakan untuk enkripsi dan dekripsi. Kecuali tugas Anda secara eksplisit memerlukan enkripsi asimetris, kunci KMS enkripsi simetris, yang tidak pernah dibiarkan AWS KMS tidak terenkripsi, adalah pilihan yang baik.

AWSlayanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Secara teknis, spesifikasi kunci untuk kunci simetris adalah SYMMETRIC_DEFAULT, penggunaan kuncinya adalah ENCRYPT_DECRYPT, dan algoritma enkripsi adalah SYMMETRIC_DEFAULT. Untuk detailnya, lihat Spesifikasi kunci SYMMETRIC_DEFAULT.

Anda dapat menggunakan kunci KMS enkripsi simetris AWS KMS untuk mengenkripsi, mendekripsi, dan mengenkripsi ulang data, dan menghasilkan kunci data dan pasangan kunci data. Anda dapat membuat kunci KMS enkripsi simetris multi-wilayah, mengimpor materi kunci Anda sendiri ke kunci KMS enkripsi simetris, dan membuat kunci KMS enkripsi simetris di toko kunci khusus. Untuk tabel yang membandingkan operasi yang dapat Anda lakukan pada kunci KMS dari berbagai jenis, lihatReferensi tipe kunci.

Tombol Asymmetric KMS

Anda dapat membuat kunci KMS asimetris di. AWS KMS Kunci KMS asimetris mewakili kunci publik yang terkait secara matematis dan private key pair. Kunci privat tidak pernah membiarkan AWS KMS tidak terenkripsi. Untuk menggunakan kunci privat, Anda harus memanggil AWS KMS. Anda dapat menggunakan kunci publik di dalam AWS KMS dengan memanggil operasi AWS KMS API, atau Anda dapat mengunduh kunci publik dan menggunakannya di luarAWS KMS. Anda juga dapat membuat kunci KMS asimetris Multi-wilayah.

Anda dapat membuat kunci KMS asimetris yang mewakili pasangan kunci RSA atau pasangan kunci SM2 (hanya Wilayah China) untuk enkripsi atau penandatanganan dan verifikasi kunci publik, atau pasangan kunci kurva elips untuk penandatanganan dan verifikasi.

Untuk informasi selengkapnya tentang membuat dan menggunakan kunci KMS asimetris, lihat. Kunci asimetris di AWS KMS

Kunci HMAC KMS

Kunci KMS HMAC mewakili kunci simetris dengan panjang yang bervariasi yang digunakan untuk menghasilkan dan memverifikasi kode otentikasi pesan berbasis hash (HMAC). Materi kunci untuk kunci HMAC tidak pernah dibiarkan tidak AWS KMS terenkripsi. Untuk menggunakan kunci HMAC, panggil operasi `GenerateMac`atau `VerifyMac`API.

Anda juga dapat membuat kunci KMS HMAC Multi-wilayah.

Untuk informasi selengkapnya tentang membuat dan menggunakan kunci HMAC KMS, lihat. Kunci HMAC di AWS KMS

Kunci data

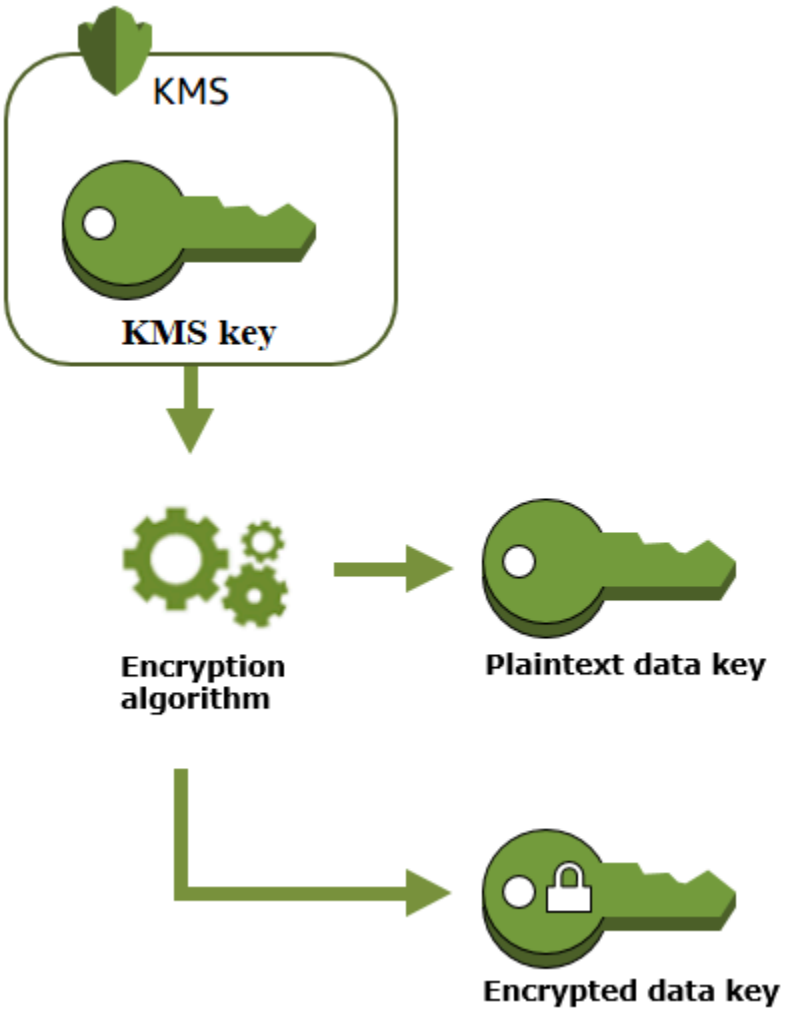
Kunci data adalah kunci simetris yang dapat Anda gunakan untuk mengenkripsi data, termasuk sejumlah besar data dan kunci enkripsi data lainnya. Tidak seperti kunci KMS simetris, yang tidak dapat diunduh, kunci data dikembalikan kepada Anda untuk digunakan di luar. AWS KMS

Saat AWS KMS menghasilkan kunci data, ia mengembalikan kunci data plaintext untuk segera digunakan (opsional) dan salinan kunci data terenkripsi yang dapat Anda simpan dengan aman bersama data. Ketika Anda siap untuk mendekripsi data, pertama-tama Anda meminta AWS KMS untuk mendekripsi kunci data terenkripsi.

AWS KMSmenghasilkan, mengenkripsi, dan mendekripsi kunci data. Namun, AWS KMS tidak menyimpan, mengelola, atau melacak kunci data Anda, atau melakukan operasi kriptografi dengan kunci data. Anda harus menggunakan dan mengelola kunci data di luar AWS KMS. Untuk bantuan menggunakan kunci data dengan aman, lihat. AWS Encryption SDK

Membuat kunci data

Untuk membuat kunci data, panggil GenerateDataKeyoperasi. AWS KMSmenghasilkan kunci data. Kemudian mengenkripsi salinan kunci data di bawah kunci KMS enkripsi simetris yang Anda tentukan. Operasi mengembalikan salinan plaintext dari kunci data dan salinan kunci data yang dienkripsi di bawah kunci KMS. Gambar berikut menunjukkan operasi ini.

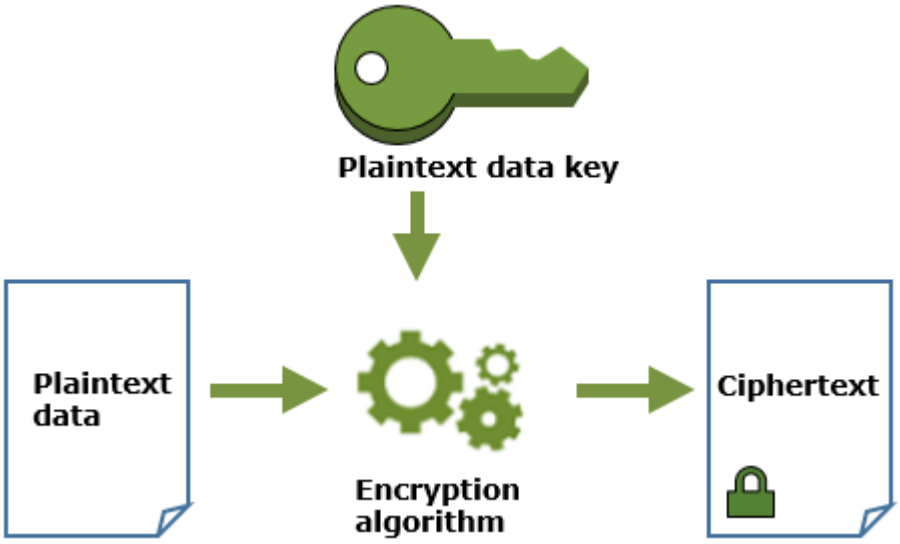


AWS KMSjuga mendukung GenerateDataKeyWithoutPlaintextoperasi, yang mengembalikan hanya kunci data terenkripsi. Jika Anda perlu menggunakan kunci data, minta AWS KMS untuk mendekripsinya.

Mengenkripsi data dengan kunci data

AWS KMS tidak dapat menggunakan kunci data untuk mengenkripsi data. Tetapi Anda bisa menggunakan kunci data di luar AWS KMS, seperti menggunakan OpenSSL atau pustaka kriptografi seperti AWS Encryption SDK.

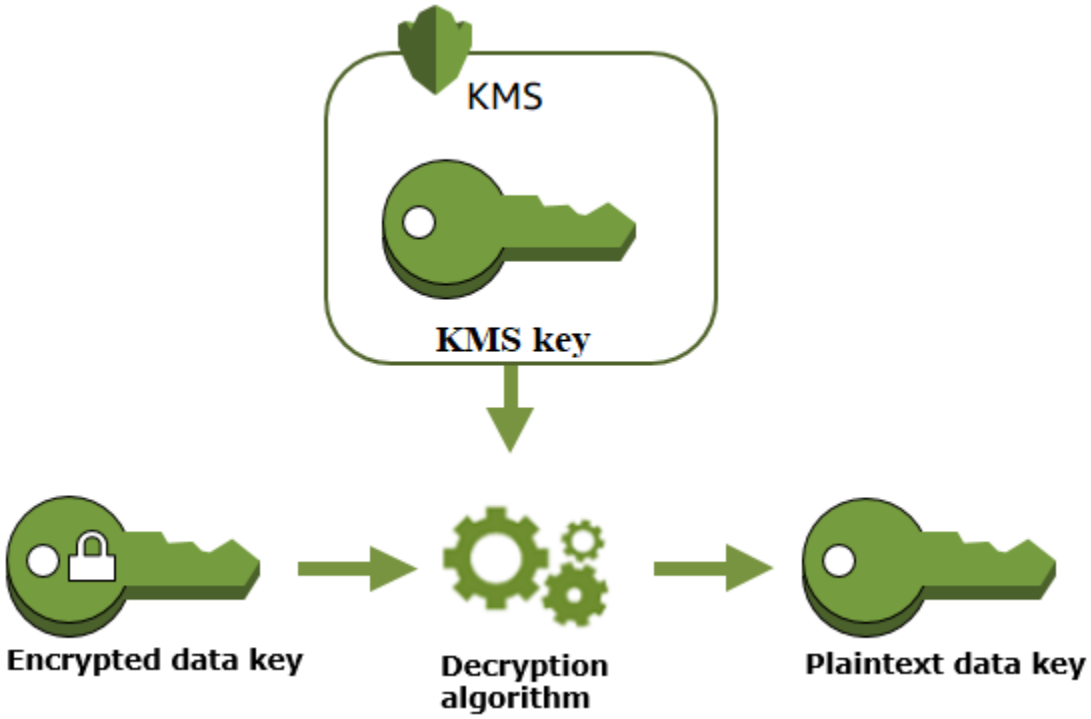
Setelah menggunakan kunci data plaintext untuk mengenkripsi data, hapus kunci data dari memori sesegera mungkin. Anda dapat dengan aman menyimpan kunci data terenkripsi dengan data terenkripsi sehingga tersedia untuk mendekripsi data.



Mendekripsi data dengan kunci data

Untuk mendekripsi data Anda, teruskan kunci data terenkripsi ke operasi Dekripsi. AWS KMSmenggunakan kunci KMS Anda untuk mendekripsi kunci data dan kemudian mengembalikan kunci data plaintext. Gunakan kunci data plaintext untuk mendekripsi data Anda kemudian hapus kunci data plaintext dari memori sesegera mungkin.

Diagram berikut menunjukkan cara menggunakan operasi `Decrypt` untuk mendekripsi kunci data terenkripsi.



Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data

Ketika kunci KMS menjadi tidak dapat digunakan, efeknya hampir seketika (tergantung pada konsistensi akhirnya). Status kunci dari perubahan kunci KMS untuk mencerminkan kondisi barunya, dan semua permintaan untuk menggunakan kunci KMS dalam operasi kriptografi gagal.

Namun, efek pada kunci data yang dienkripsi oleh kunci KMS, dan pada data yang dienkripsi oleh kunci data, ditunda hingga kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data.

Kunci KMS dapat menjadi tidak dapat digunakan karena berbagai alasan, termasuk tindakan berikut yang mungkin Anda lakukan.

- Menonaktifkan tombol KMS
- Menjadwalkan kunci KMS untuk dihapus
- Menghapus materi kunci dari kunci KMS dengan bahan kunci impor, atau membiarkan bahan kunci yang diimpor kedaluwarsa.
- Memutuskan sambungan penyimpanan AWS CloudHSM kunci yang menghosting kunci KMS, atau menghapus kunci dari AWS CloudHSM cluster yang berfungsi sebagai bahan kunci untuk kunci KMS.

- Memutuskan sambungan penyimpanan kunci eksternal yang menghosting kunci KMS, atau tindakan lain yang mengganggu permintaan enkripsi dan dekripsi ke proxy penyimpanan kunci eksternal, termasuk menghapus kunci eksternal dari pengelola kunci eksternalnya.

Efek ini sangat penting bagi banyak orang Layanan AWS yang menggunakan kunci data untuk melindungi sumber daya yang dikelola layanan. Contoh berikut menggunakan Amazon Elastic Block Store (Amazon EBS) dan Amazon Elastic Compute Cloud (Amazon EC2). Berbeda Layanan AWS menggunakan kunci data dengan cara yang berbeda. Untuk detailnya, lihat bagian Perlindungan data pada bagian Keamanan untuk bagianLayanan AWS.

Misalnya, pertimbangkan skenario ini:

1. Anda membuat volume EBS terenkripsi dan menentukan kunci KMS untuk melindunginya. Amazon EBS meminta AWS KMS untuk menggunakan kunci KMS Anda untuk menghasilkan kunci data terenkripsi untuk volume. Amazon EBS menyimpan kunci data terenkripsi dengan metadata volume.
2. Saat Anda melampirkan volume EBS ke instans EC2, Amazon EC2 menggunakan kunci KMS Anda untuk mendekripsi kunci data terenkripsi volume EBS. Amazon EC2 menggunakan kunci data di perangkat keras Nitro, yang bertanggung jawab untuk mengenkripsi semua disk I/O ke volume EBS. Kunci data tetap ada di perangkat keras Nitro sementara volume EBS dilampirkan ke instans EC2.
3. Anda melakukan tindakan yang membuat kunci KMS tidak dapat digunakan. Ini tidak memiliki efek langsung pada instans EC2 atau volume EBS. Amazon EC2 menggunakan kunci data—bukan kunci KMS—untuk mengenkripsi semua disk I/O saat volume dilampirkan ke instans.
4. Namun, saat volume EBS terenkripsi dicopot dari instans EC2, Amazon EBS menghapus kunci data dari perangkat keras Nitro. Pada saat volume EBS yang terenkripsi dilampirkan ke instans EC2, pelampiran akan gagal karena Amazon EBS tidak dapat menggunakan kunci KMS untuk mendekripsi kunci data terenkripsi dari volume tersebut. Untuk menggunakan volume EBS lagi, Anda harus membuat kunci KMS dapat digunakan lagi.

Pasangan kunci data

Pasangan kunci data adalah kunci data asimetris yang terdiri dari kunci publik dan kunci pribadi yang terkait secara matematis. Mereka dirancang untuk digunakan dalam enkripsi sisi klien dan dekripsi atau penandatanganan dan verifikasi di luar. AWS KMS

Berbeda dengan pasangan kunci data yang dihasilkan alat seperti OpenSSLAWS KMS, melindungi kunci pribadi di setiap data key pair di bawah kunci KMS enkripsi simetris yang Anda tentukan. AWS KMS Namun, AWS KMS tidak menyimpan, mengelola, atau melacak pasangan kunci data Anda, atau melakukan operasi kriptografi dengan pasangan kunci data. Anda harus menggunakan dan mengelola pasangan kunci data di luar AWS KMS.

AWS KMS mendukung jenis pasangan kunci data berikut:

- Pasangan kunci RSA: RSA_2048, RSA_3072, dan RSA_4096
- Pasangan kunci kurva elips: ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521, dan ECC_SECG_P256K1
- Pasangan kunci SM (hanya Wilayah China): SM2

Jenis pasangan kunci data yang Anda pilih biasanya bergantung pada kasus penggunaan atau persyaratan peraturan Anda. Sebagian besar sertifikat memerlukan kunci RSA. Kunci kurva elips sering digunakan untuk tanda tangan digital. Kunci ECC_SECG_P256K1 umumnya digunakan untuk cryptocurrency. AWS KMSmerekomendasikan agar Anda menggunakan pasangan kunci ECC untuk menandatangani, dan menggunakan pasangan kunci RSA untuk enkripsi atau penandatanganan, tetapi tidak keduanya. Namun, AWS KMS tidak dapat menerapkan pembatasan apa pun pada penggunaan pasangan kunci data di luar AWS KMS.

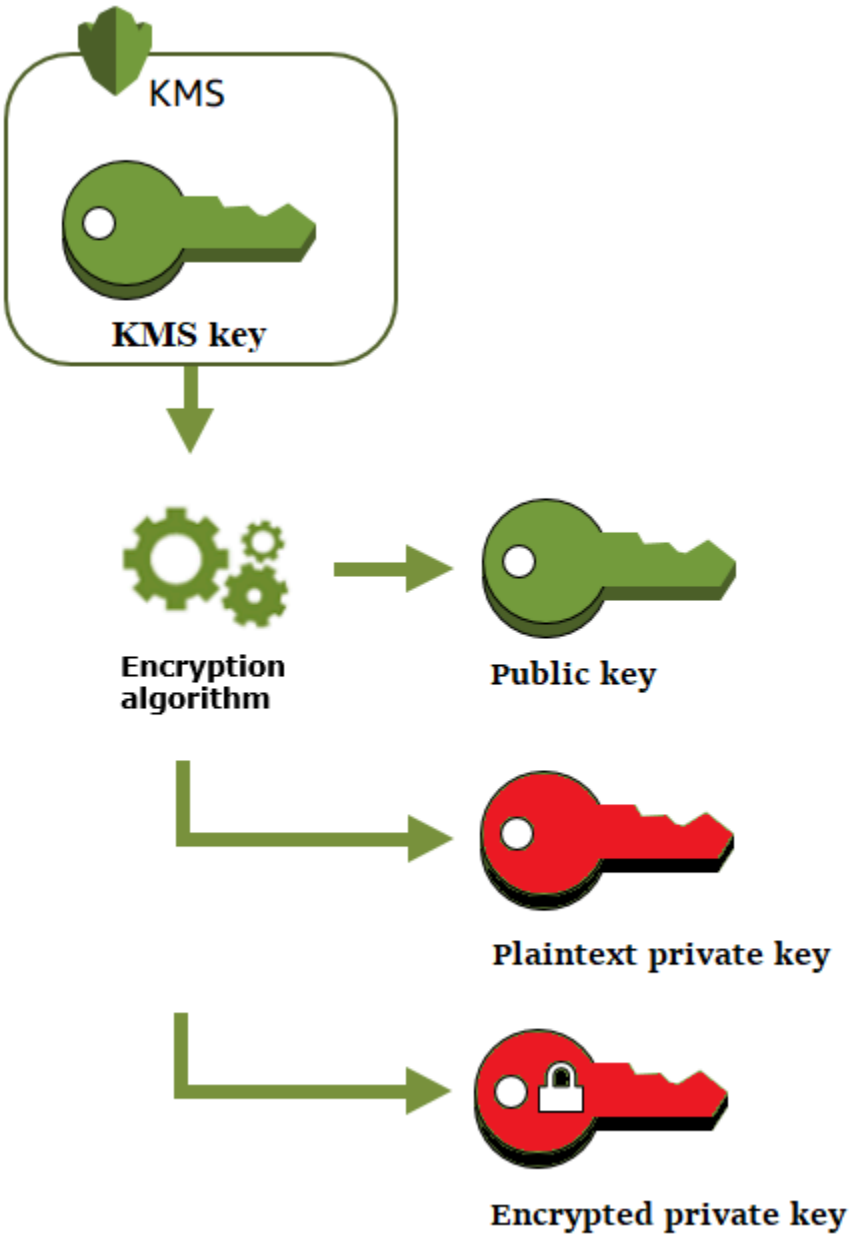
Membuat pasangan kunci data

Untuk membuat data key pair, panggil GenerateDataKeyPairor GenerateDataKeyPairWithoutPlaintextoperations. Tentukan kunci KMS enkripsi simetris yang ingin Anda gunakan untuk mengenkripsi kunci pribadi.

`GenerateDataKeyPair` mengembalikan kunci publik plaintext, kunci privat plaintext, dan kunci privat terenkripsi. Gunakan operasi ini ketika Anda memerlukan kunci privat plaintext segera, seperti untuk menghasilkan tanda tangan digital.

`GenerateDataKeyPairWithoutPlaintext` mengembalikan kunci publik plaintext dan kunci privat terenkripsi, tetapi bukan kunci privat plaintext. Gunakan operasi ini ketika Anda tidak memerlukan kunci privat plaintext segera, seperti ketika Anda mengenkripsi dengan kunci publik. Kemudian, ketika Anda membutuhkan kunci privat plaintext untuk mendekripsi data, Anda dapat memanggil operasi Decrypt.

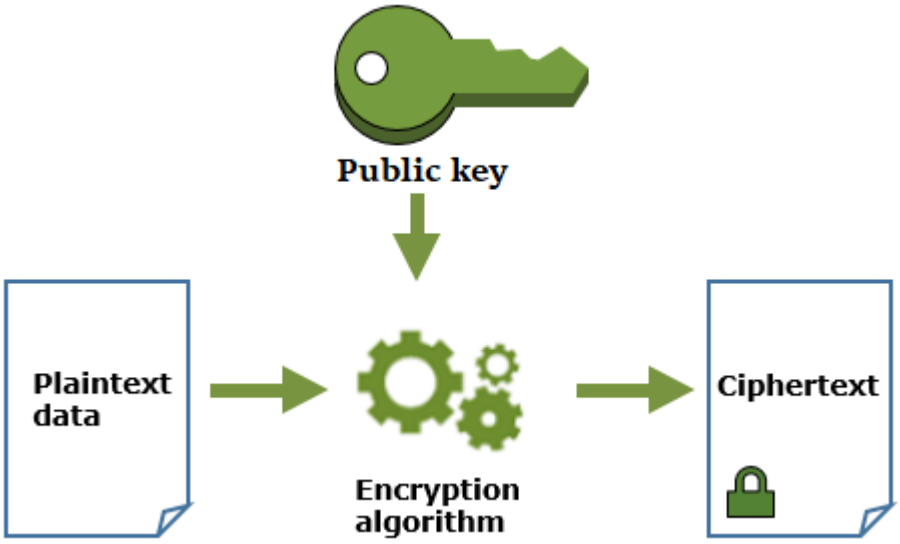
Gambar berikut menunjukkan operasi `GenerateDataKeyPair`. Operasi `GenerateDataKeyPairWithoutPlaintext` menghilangkan kunci privat plaintext.



Mengenkripsi data dengan pasangan kunci data

Ketika Anda mengenkripsi dengan pasangan kunci data, Anda menggunakan kunci publik pasangan untuk mengenkripsi data dan kunci privat pasangan yang sama untuk mendekripsi data. Biasanya, Anda menggunakan pasangan kunci data ketika banyak pihak perlu mengenkripsi data yang hanya pihak dengan kunci pribadi dapat mendekripsi.

Pihak dengan kunci publik menggunakan kunci tersebut untuk mengenkripsi data, seperti yang ditunjukkan dalam diagram berikut.

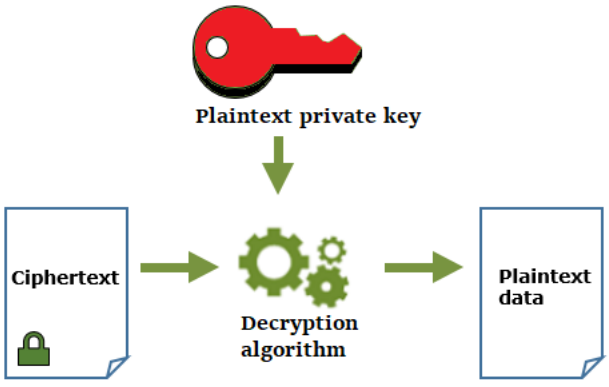


Mendekripsi data dengan pasangan kunci data

Untuk mendekripsi data Anda, gunakan kunci privat dalam pasangan kunci data. Agar operasi berhasil, kunci publik dan privat harus berasal dari pasangan kunci data yang sama, dan Anda harus menggunakan algoritme enkripsi yang sama.

Untuk mendekripsi kunci privat terenkripsi, berikan kunci untuk operasi Decrypt. Gunakan kunci privat plaintext untuk mendekripsi data. Kemudian, hapus kunci privat plaintext dari memori sesegera mungkin.

Diagram berikut menunjukkan cara menggunakan kunci privat dalam pasangan kunci data untuk mendekripsi ciphertext.



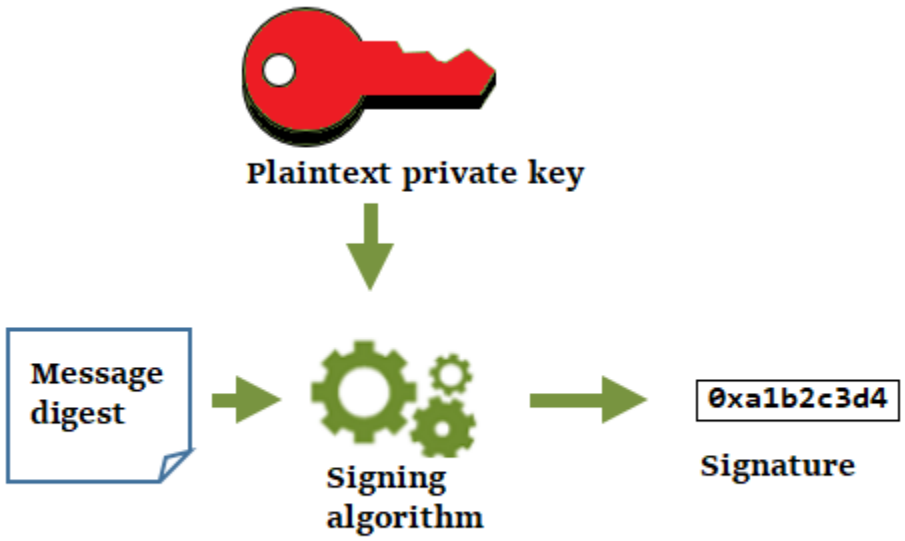
Menandatangani pesan dengan pasangan kunci data

Untuk menghasilkan tanda tangan kriptografi untuk suatu pesan, gunakan kunci privat dalam pasangan kunci data. Siapa pun yang memiliki kunci publik dapat menggunakannya untuk memverifikasi bahwa pesan telah ditandatangani dengan kunci privat Anda dan bahwa pesan tersebut tidak berubah sejak ditandatangani.

Jika Anda mengenkripsi kunci pribadi Anda, teruskan kunci pribadi terenkripsi ke operasi Dekripsi. AWS KMSmenggunakan kunci KMS Anda untuk mendekripsi kunci data dan kemudian mengembalikan kunci pribadi plaintext. Gunakan kunci privat plaintext untuk menghasilkan tanda tangan. Kemudian, hapus kunci privat plaintext dari memori sesegera mungkin.

Untuk menandatangani pesan, buat intisari pesan menggunakan fungsi hash kriptografi, seperti perintah `dgst`di OpenSSL. Kemudian, berikan kunci privat plaintext Anda untuk algoritme penandatanganan. Hasilnya adalah tanda tangan yang mewakili isi pesan. (Anda mungkin dapat menandatangani pesan yang lebih singkat tanpa terlebih dahulu membuat intisari. Ukuran pesan maksimum bervariasi dengan alat penandatanganan yang Anda gunakan.)

Diagram berikut menunjukkan cara menggunakan kunci privat dalam pasangan kunci data untuk menandatangani pesan.

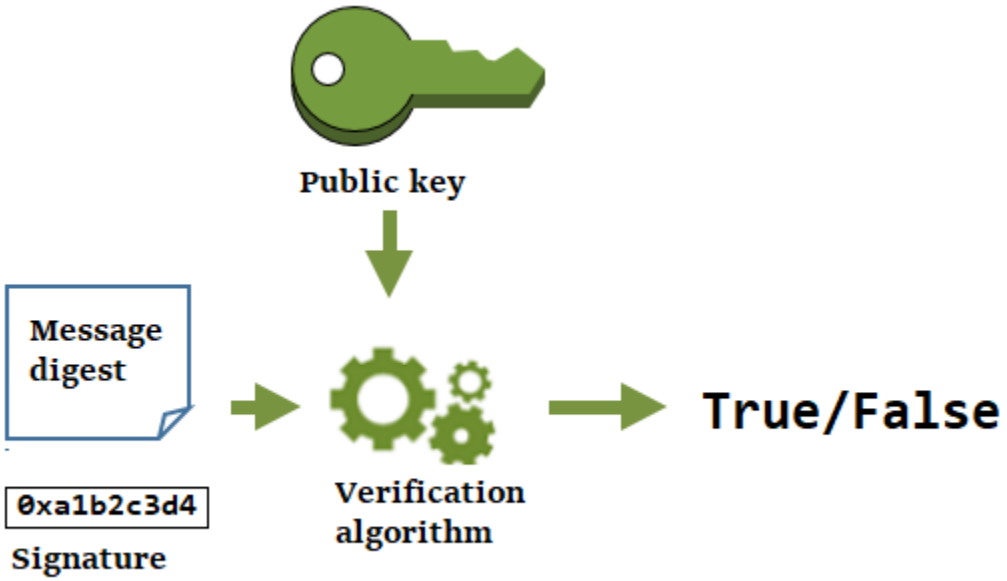


Memverifikasi tanda tangan dengan pasangan kunci data

Siapa pun yang memiliki kunci publik dalam pasangan kunci data Anda dapat menggunakannya untuk memverifikasi tanda tangan yang Anda hasilkan dengan kunci privat Anda. Verifikasi mengonfirmasi bahwa pengguna yang diotorisasi menandatangani pesan tersebut dengan kunci privat dan algoritme penandatanganan yang ditentukan, dan pesan tersebut tidak berubah sejak ditandatangani.

Agar berhasil, pihak yang memverifikasi tanda tangan harus menghasilkan jenis intisari yang sama, menggunakan algoritme yang sama, dan menggunakan kunci publik yang sesuai dengan kunci privat yang digunakan untuk menandatangani pesan.

Diagram berikut menunjukkan cara menggunakan kunci publik dalam pasangan kunci data untuk memverifikasi pesan.



Alias

Gunakan alias sebagai nama ramah untuk kunci KMS. Misalnya, Anda dapat merujuk ke kunci KMS sebagai kunci uji, bukan 1234abcd-12ab-34cd-56ef-1234567890ab.

Alias membuatnya lebih mudah untuk mengidentifikasi kunci KMS di. AWS Management Console Anda dapat menggunakan alias untuk mengidentifikasi kunci KMS dalam beberapa AWS KMS operasi, termasuk operasi kriptografi. Dalam aplikasi, Anda dapat menggunakan satu alias untuk merujuk ke kunci KMS yang berbeda di masing-masing. Wilayah AWS

Anda juga dapat mengizinkan dan menolak akses ke kunci KMS berdasarkan aliasnya tanpa mengedit kebijakan atau mengelola hibah. Fitur ini adalah bagian dari dukungan AWS KMS untuk kontrol akses berbasis atribut (ABAC). Untuk rincian selengkapnya, lihat ABAC untuk AWS KMS.

DalamAWS KMS, alias adalah sumber daya independen, bukan properti kunci KMS. Dengan demikian, Anda dapat menambahkan, mengubah, dan menghapus alias tanpa mempengaruhi kunci KMS terkait.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam nama alias. Alias dapat muncul dalam plaintext di CloudTrail log dan output lainnya.

Pelajari lebih lanjut:

- Untuk informasi mendetail tentang alias, lihat Menggunakan alias.
- Untuk informasi tentang format pengidentifikasi kunci, termasuk alias, lihat Pengidentifikasi kunci () KeyId.
- Untuk bantuan menemukan alias yang terkait dengan kunci KMS, lihat Menemukan nama alias dan ARN alias
- Untuk contoh membuat dan mengelola alias dalam beberapa bahasa pemrograman, lihat Bekerja dengan alias.

Penyimpanan kunci kustom

Toko kunci khusus adalah AWS KMS sumber daya yang didukung oleh manajer kunci di luar AWS KMS yang Anda miliki dan kelola. Ketika Anda menggunakan kunci KMS di toko kunci khusus untuk operasi kriptografi, operasi kriptografi sebenarnya dilakukan di manajer kunci Anda menggunakan kunci kriptografinya.

AWS KMSmendukung penyimpanan AWS CloudHSM kunci yang didukung oleh AWS CloudHSM cluster dan penyimpanan kunci eksternal yang didukung oleh manajer kunci eksternal di luarAWS.

Untuk informasi selengkapnya, lihat Penyimpanan kunci kustom.

Operasi kriptografi

DalamAWS KMS, operasi kriptografi adalah operasi API yang menggunakan kunci KMS untuk melindungi data. Karena kunci KMS tetap berada di dalamAWS KMS, Anda harus memanggil AWS KMS untuk menggunakan kunci KMS dalam operasi kriptografi.

Untuk melakukan operasi kriptografi dengan kunci KMS, gunakan AWS SDK, AWS Command Line Interface (AWS CLI), atau. AWS Tools for PowerShell Anda tidak dapat melakukan operasi kriptografi di konsol AWS KMS. Untuk contoh memanggil operasi kriptografi dalam beberapa bahasa pemrograman, lihat Memprogram API AWS KMS.

Tabel berikut mencantumkan operasi kriptografis AWS KMS. Ini juga menunjukkan jenis kunci dan persyaratan penggunaan kunci untuk kunci KMS yang digunakan dalam operasi.

Operasi	Tipe Kunci	Penggunaan kunci
Dekripsi	Simetris atau asimetris	`ENCRYPT_DECRYPT`
Enkripsi	Simetris atau asimetris	`ENCRYPT_DECRYPT`
GenerateDataKey	Simetris	`ENCRYPT_DECRYPT`
GenerateDataKeyPair	Simetris [1] Tidak didukung pada kunci KMS di toko kunci khusus.	`ENCRYPT_DECRYPT`
GenerateDataKeyPairWithoutPlaintext	Simetris [1] Tidak didukung pada kunci KMS di toko kunci khusus.	`ENCRYPT_DECRYPT`
GenerateDataKeyWithoutPlaintext	Simetris	`ENCRYPT_DECRYPT`
GenerateMac	HMAC	`GENERATE_VERIFY_MAC`

Operasi	Tipe Kunci	Penggunaan kunci
GenerateRandom	N/A. Operasi ini tidak menggunakan kunci KMS.	N/A
ReEncrypt	Simetris atau asimetris	`ENCRYPT_DECRYPT`
Tanda	Asimetris	`SIGN_VERIFY`
Verifikasi	Asimetris	`SIGN_VERIFY`
VerifyMac	HMAC	`GENERATE_VERIFY_MAC`

[1] Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.

Untuk informasi tentang izin untuk operasi kriptografi, lihat the section called “Referensi izin”.

Untuk membuat AWS KMS responsif dan sangat fungsional untuk semua pengguna, AWS KMS menetapkan kuota pada jumlah operasi kriptografi yang disebut dalam setiap detik. Untuk detailnya, lihat the section called “Kuota bersama untuk operasi kriptografis”.

Pengidentifikasi kunci () KeyId

Pengidentifikasi kunci bertindak seperti nama untuk kunci KMS Anda. Mereka membantu Anda mengenali kunci KMS Anda di konsol. Anda menggunakannya untuk menunjukkan kunci KMS mana yang ingin Anda gunakan dalam operasi AWS KMS API, kebijakan utama, kebijakan IAM, dan hibah. Nilai pengidentifikasi kunci sama sekali tidak terkait dengan materi kunci yang terkait dengan kunci KMS.

AWS KMS menentukan beberapa pengidentifikasi kunci. Ketika Anda membuat kunci KMS, AWS KMS menghasilkan kunci ARN dan kunci ID, yang merupakan properti dari kunci KMS. Saat Anda membuat alias, AWS KMS menghasilkan alias ARN berdasarkan nama alias yang Anda tentukan. Anda dapat melihat pengidentifikasi kunci dan alias dalam AWS Management Console dan dalam AWS KMS API.

Di AWS KMS konsol, Anda dapat melihat dan memfilter kunci KMS berdasarkan ARN kunci mereka, ID kunci, atau nama alias, dan mengurutkan berdasarkan ID kunci dan nama alias. Untuk bantuan menemukan pengenal kunci di konsol, lihatthe section called “Menemukan ID kunci dan kunci ARN”.

Di AWS KMS API, parameter yang Anda gunakan untuk mengidentifikasi kunci KMS diberi nama `KeyId` atau variasi, seperti `TargetKeyId` atau`DestinationKeyId`. Namun, nilai-nilai parameter tersebut tidak terbatas pada ID kunci. Beberapa dapat mengambil pengidentifikasi kunci yang valid. Untuk informasi tentang nilai untuk setiap parameter, lihat deskripsi parameter di Referensi AWS Key Management Service API.

> **Note**
>
> Saat menggunakan AWS KMS API, hati-hati terkait pengidentifikasi kunci yang Anda gunakan. API yang berbeda memerlukan pengidentifikasi kunci yang berbeda. Secara umum, gunakan pengenal kunci paling lengkap dan praktis untuk tugas Anda.

AWS KMS mendukung pengidentifikasi kunci berikut.

ARN kunci

Kunci ARN adalah Nama Sumber Daya Amazon (ARN) dari kunci KMS. Ini adalah pengidentifikasi unik dan sepenuhnya memenuhi syarat untuk kunci KMS. Sebuah ARN kunci termasuk Akun AWS, Wilayah, dan ID kunci. Untuk bantuan menemukan ARN kunci dari kunci KMS, lihat. the section called “Menemukan ID kunci dan kunci ARN”

Format ARN kunci adalah sebagai berikut:

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

Berikut ini adalah contoh kunci ARN untuk kunci KMS Single-region.

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

Elemen *`key-id`* dari ARN kunci dari kunci multi-Wilayah dimulai dengan prefiks `mrk-`. Berikut ini adalah contoh kunci ARN untuk kunci Multi-region.

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

ID Kunci

ID kunci secara unik mengidentifikasi kunci KMS dalam akun dan Wilayah. Untuk bantuan menemukan ID kunci dari kunci KMS, lihatthe section called “Menemukan ID kunci dan kunci ARN”.

Berikut ini adalah ID kunci contoh untuk kunci KMS Single-region.

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

ID kunci dari kunci multi-Wilayah dimulai dengan prefiks `mrk-`. Berikut ini adalah ID kunci contoh untuk kunci Multi-region.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN Alias

ARN alias adalah Amazon Resource Name (ARN) dari alias AWS KMS. Ini adalah pengidentifikasi unik yang sepenuhnya memenuhi syarat untuk alias, dan untuk kunci KMS yang diwakilinya. ARN alias termasuk Akun AWS, Wilayah, dan nama alias.

Pada waktu tertentu, alias ARN mengidentifikasi satu kunci KMS tertentu. Namun, karena Anda dapat mengubah kunci KMS yang terkait dengan alias, alias ARN dapat mengidentifikasi kunci KMS yang berbeda pada waktu yang berbeda. Untuk bantuan menemukan alias ARN dari kunci KMS, lihat. Menemukan nama alias dan ARN alias

Format ARN alias adalah sebagai berikut:

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

Berikut ini adalah ARN alias untuk `ExampleAlias` fiktif.

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

Nama alias

Nama alias adalah satu string berisi hingga 256 karakter. Ini secara unik mengidentifikasi kunci KMS terkait dalam akun dan Wilayah. Dalam AWS KMS API, nama alias selalu diawali dengan

`alias/`. Untuk bantuan menemukan nama alias kunci KMS, lihat. Menemukan nama alias dan ARN alias

Format nama alias adalah sebagai berikut:

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

Sebagai contoh:

```
alias/ExampleAlias
```

`aws/`Awalan untuk nama alias dicadangkan untuk. Kunci yang dikelola AWS Anda tidak dapat membuat alias dengan prefiks ini. Misalnya, nama alias Kunci yang dikelola AWS untuk Amazon Simple Storage Service (Amazon S3) Simple Storage Service S3) adalah sebagai berikut.

```
alias/aws/s3
```

Material kunci

Materi kunci adalah string bit yang digunakan dalam algoritma kriptografi. Materi kunci rahasia harus dirahasiakan untuk melindungi operasi kriptografi yang menggunakannya. Materi kunci publik dirancang untuk dibagikan.

Setiap kunci KMS menyertakan referensi ke materi utamanya dalam metadatanya. Asal material kunci kunci KMS enkripsi simetris dapat bervariasi. Anda dapat menggunakan material kunci yang dihasilkan AWS KMS, material kunci yang dihasilkan dalam klaster AWS CloudHSM dari penyimpanan kunci kustom, atau impor material kunci Anda sendiri. Jika Anda menggunakan bahan AWS KMS kunci untuk kunci KMS enkripsi simetris Anda, Anda dapat mengaktifkan rotasi otomatis materi kunci Anda.

Secara default, setiap kunci KMS memiliki bahan kunci yang unik. Namun, Anda dapat membuat satu set kunci multi-wilayah dengan material kunci yang sama.

Asal material kunci

Asal material utama adalah properti kunci KMS yang mengidentifikasi sumber bahan kunci dalam kunci KMS. Anda memilih asal bahan utama saat Anda membuat kunci KMS, dan Anda tidak dapat

mengubahnya. Sumber material utama mempengaruhi keamanan, daya tahan, ketersediaan, latensi, dan karakteristik throughput kunci KMS.

Untuk menemukan asal material kunci dari kunci KMS, gunakan DescribeKeyoperasi, atau lihat nilai Origin pada tab konfigurasi Cryptographic pada halaman detail untuk kunci KMS di konsol. AWS KMS Untuk bantuan, lihat Melihat Kunci.

Kunci KMS dapat memiliki salah satu nilai asal material utama berikut.

`AWS_KMS`

AWS KMSmembuat dan mengelola materi kunci untuk kunci KMS di toko kuncinya sendiri. Ini adalah nilai default dan yang direkomendasikan untuk sebagian besar kunci KMS.

Untuk bantuan dalam membuat kunci dengan material kunci dari AWS KMS, lihat Membuat kunci.

`EXTERNAL (Import key material)`

Kunci KMS telah mengimpor bahan kunci. Saat Anda membuat kunci KMS dengan asal material `External` kunci, kunci KMS tidak memiliki materi kunci. Nanti, Anda dapat mengimpor materi kunci ke kunci KMS. Jika Anda menggunakan material kunci impor, Anda perlu mengamankan dan mengelola material kunci tersebut di luar AWS KMS, termasuk mengganti material kunci jika masa berlakunya berakhir. Untuk detail, lihat Tentang material kunci yang diimpor.

Untuk bantuan membuat kunci KMS untuk materi kunci impor, lihatLangkah 1: Buat kunci KMS tanpa bahan kunci.

`AWS_CLOUDHSM`

AWS KMSmembuat materi kunci di AWS CloudHSM cluster untuk toko AWS CloudHSM kunci Anda.

Untuk bantuan membuat kunci KMS di toko AWS CloudHSM kunci, lihatMembuat kunci KMS di toko AWS CloudHSM kunci.

`EXTERNAL_KEY_STORE`

Materi utamanya adalah kunci kriptografi di manajer kunci eksternal di luar. AWS Asal ini hanya didukung untuk kunci KMS di toko kunci eksternal.

Untuk bantuan membuat kunci KMS di toko kunci eksternal, lihatMembuat kunci KMS di toko kunci eksternal.

Spesifikasi kunci

Spesifikasi kunci adalah properti yang mewakili konfigurasi kriptografi kunci. Arti dari spesifikasi kunci berbeda dengan tipe kunci.

- AWS KMSkunci — Spesifikasi kunci menentukan apakah kunci KMS simetris atau asimetris. Ini juga menentukan jenis bahan utamanya, dan algoritma yang didukungnya. Anda memilih spesifikasi kunci ketika Anda membuat kunci KMS, dan Anda tidak dapat mengubahnya. Spesifikasi kunci default, SYMMETRIC_DEFAULT, mewakili kunci enkripsi simetris 256-bit.

  > **Note**
  >
  > Kunci `KeySpec` untuk KMS dikenal sebagai a`CustomerMasterKeySpec`. `CustomerMasterKeySpec`Parameter CreateKeyoperasi tidak digunakan lagi. Sebagai gantinya, gunakan `KeySpec` parameter, yang bekerja dengan cara yang sama. Untuk mencegah perubahan yang melanggar, respons `CreateKey` dan DescribeKeyoperasi sekarang mencakup keduanya `KeySpec` dan `CustomerMasterKeySpec` anggota dengan nilai yang sama.

  Untuk daftar spesifikasi kunci dan membantu memilih spesifikasi kunci, lihat Memilih spesifikasi kunci. Untuk menemukan spesifikasi kunci kunci KMS, gunakan DescribeKeyoperasi, atau lihat tab konfigurasi Kriptografi pada halaman detail untuk kunci KMS di konsol. AWS KMS Untuk bantuan, lihat Melihat Kunci.

  Untuk membatasi spesifikasi kunci yang dapat digunakan prinsipal saat membuat kunci KMS, gunakan kunci kondisi kms:. KeySpec Anda juga dapat menggunakan tombol `kms:KeySpec` kondisi untuk mengizinkan prinsipal memanggil AWS KMS operasi hanya pada kunci KMS dengan spesifikasi kunci tertentu. Misalnya, Anda dapat menolak izin untuk menjadwalkan penghapusan kunci KMS apa pun dengan spesifikasi kunci. `RSA_4096`

- Data keys (GenerateDataKey) — Spesifikasi kunci menentukan panjang kunci data AES.
- Data keys pairs (GenerateDataKeyPair) — Spesifikasi key pair menentukan jenis material kunci dalam data key pair.

Penggunaan kunci

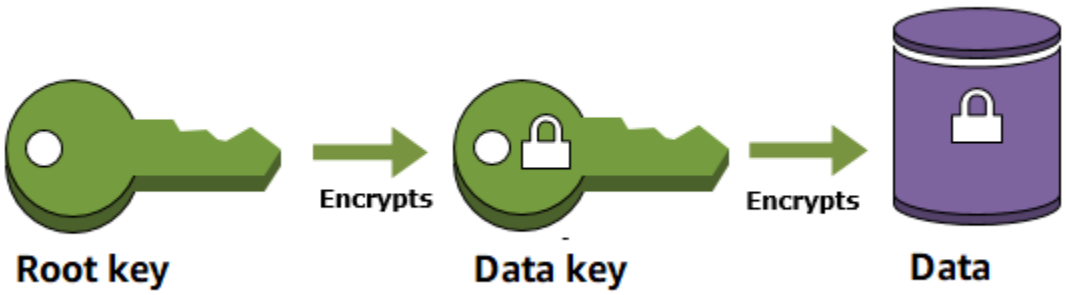
Penggunaan kunci adalah properti yang menentukan operasi kriptografi yang didukung kunci. Kunci KMS dapat memiliki penggunaan kunci`ENCRYPT_DECRYPT`,`SIGN_VERIFY`, atau`GENERATE_VERIFY_MAC`. Setiap kunci KMS hanya dapat memiliki satu penggunaan kunci. Menggunakan kunci KMS untuk lebih dari satu jenis operasi membuat produk dari kedua operasi lebih rentan terhadap serangan.

Untuk bantuan memilih penggunaan kunci untuk kunci KMS Anda, lihatMemilih penggunaan kunci. Untuk menemukan penggunaan kunci kunci KMS, gunakan DescribeKeyoperasi, atau pilih tab konfigurasi Kriptografi pada halaman detail untuk kunci KMS di konsol. AWS KMS Untuk bantuan, lihat Melihat Kunci.

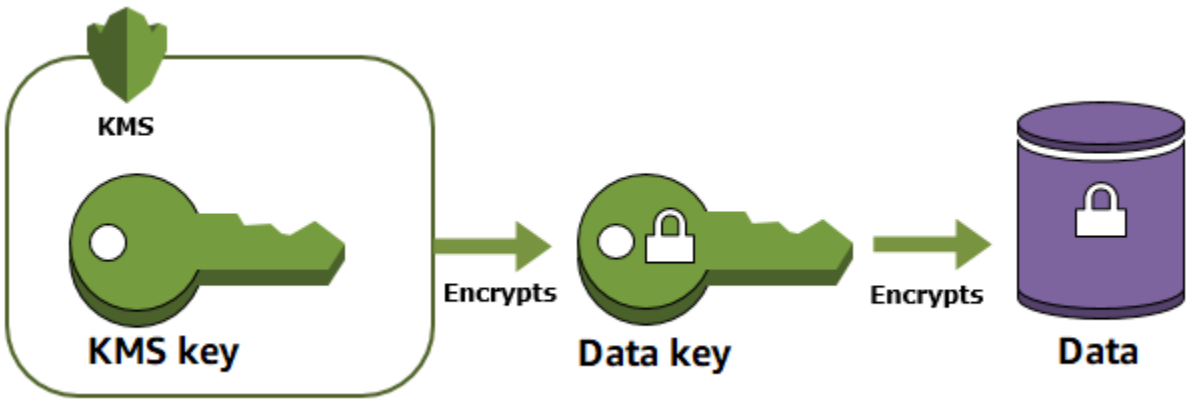
Enkripsi amplop

Saat Anda mengenkripsi data, data Anda terlindungi, tetapi Anda harus melindungi kunci enkripsi. Salah satu strateginya adalah dengan mengenkripsikannya. Enkripsi amplop adalah praktik mengenkripsi data plaintext dengan kunci data, kemudian mengenkripsi kunci data di bawah kunci lain.

Anda bahkan dapat mengenkripsi kunci enkripsi data di bawah kunci enkripsi lain, dan mengenkripsi kunci enkripsi tersebut di bawah kunci enkripsi lain. Namun pada akhirnya, satu kunci harus tetap dalam plaintext sehingga Anda dapat mendekripsi kunci dan data Anda. Kunci enkripsi kunci plaintext tingkat atas ini dikenal sebagai kunci root.



AWS KMSmembantu Anda melindungi kunci enkripsi Anda dengan menyimpan dan mengelolanya dengan aman. Kunci root yang disimpan diAWS KMS, yang dikenal sebagai AWS KMS keys, tidak pernah membiarkan modul keamanan perangkat keras yang divalidasi AWS KMS FIPS tidak terenkripsi. Untuk menggunakan kunci KMS, Anda harus meneleponAWS KMS.



Enkripsi amplop menawarkan beberapa manfaat:

- Melindungi kunci data

  Saat mengenkripsi kunci data, Anda tidak perlu khawatir menyimpan kunci data terenkripsi, karena kunci data terlindung secara inheren oleh enkripsi. Anda dapat menyimpan kunci data terenkripsi dengan aman di samping data yang dienkripsi.
- Mengenkripsi data yang sama di bawah beberapa kunci

  Operasi enkripsi dapat sangat memakan waktu, terutama ketika data yang dienkripsi adalah objek berukuran besar. Alih-alih mengenkripsi ulang data mentah beberapa kali dengan kunci yang berbeda, Anda dapat mengenkripsi ulang hanya kunci data yang melindungi data mentah.
- Menggabungkan kekuatan beberapa algoritma

  Secara umum, algoritme kunci simetris lebih cepat dan menghasilkan ciphertext yang lebih kecil dari algoritme kunci publik. Namun algoritme kunci publik memberikan pemisahan peran yang melekat dan manajemen kunci yang lebih mudah. Enkripsi amplop memungkinkan Anda menggabungkan kekuatan masing-masing strategi.

Konteks enkripsi

Semua operasi AWS KMS kriptografi dengan kunci KMS enkripsi simetris menerima konteks enkripsi, satu set opsional pasangan nilai kunci non-rahasia yang dapat berisi informasi kontekstual tambahan tentang data. AWS KMSmenggunakan konteks enkripsi sebagai data otentikasi tambahan (AAD) untuk mendukung enkripsi yang diautentikasi.

Ketika Anda menyertakan konteks enkripsi dalam permintaan enkripsi, itu terikat pada ciphertext secara kriptografis sehingga konteks enkripsi yang sama diperlukan untuk mendekripsi (atau mendekripsi dan mengenkripsi ulang) data. Jika konteks enkripsi yang disediakan dalam permintaan dekripsi tidak tepat, cocok huruf besar/kecil, permintaan dekripsi gagal. Hanya urutan pasangan kunci-nilai dalam konteks enkripsi yang dapat bervariasi.

> **Note**
>
> Anda tidak dapat menentukan konteks enkripsi dalam operasi kriptografi dengan kunci KMS asimetris atau kunci KMS HMAC. Algoritma asimetris dan algoritma MAC tidak mendukung konteks enkripsi.

Konteks enkripsi tidak rahasia dan tidak dienkripsi. Konteks enkripsi muncul dalam plaintext di Log AWS CloudTrail, sehingga Anda dapat menggunakannya untuk mengidentifikasi dan mengategorikan operasi kriptografi Anda. Konteks enkripsi Anda tidak boleh menyertakan informasi sensitif. Sebaiknya konteks enkripsi Anda menjelaskan data yang dienkripsi atau didekripsi. Misalnya, ketika mengenkripsi file, Anda mungkin menggunakan bagian dari jalur file sebagai konteks enkripsi.

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

Misalnya, saat mengenkripsi volume dan snapshot yang dibuat dengan operasi Amazon Elastic Block Store (Amazon EBS) CreateSnapshot, Amazon EBS menggunakan ID volume sebagai nilai konteks enkripsi.

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

Anda juga dapat menggunakan konteks enkripsi untuk memperbaiki atau membatasi akses ke AWS KMS keys akun Anda. Anda dapat menggunakan konteks enkripsi sebagai batasan dalam hibah dan sebagai kondisi dalam pernyataan kebijakan.

Untuk mempelajari cara menggunakan konteks enkripsi untuk melindungi integritas data terenkripsi, lihat posting Cara Melindungi Integritas Data Terenkripsi Anda dengan Menggunakan AWS Key Management Service dan EncryptionContext di Blog Keamanan. AWS

Lebih lanjut tentang konteks enkripsi.

Aturan konteks enkripsi

AWS KMSmemberlakukan aturan berikut untuk kunci dan nilai konteks enkripsi.

- Kunci dan nilai dalam pasangan konteks enkripsi harus berupa string literal sederhana. Jika Anda menggunakan jenis yang berbeda, seperti bilangan bulat atau float, AWS KMS menafsirkannya sebagai string.
- Kunci dan nilai dalam konteks enkripsi dapat mencakup karakter Unicode. Jika konteks enkripsi menyertakan karakter yang tidak diizinkan dalam kebijakan utama atau kebijakan IAM, Anda tidak akan dapat menentukan konteks enkripsi dalam kunci kondisi kebijakan, seperti `kms:EncryptionContext:`*`context-key`*dan `kms:EncryptionContextKeys`. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama. Untuk detail tentang aturan dokumen kebijakan IAM, lihat Persyaratan nama IAM di Panduan Pengguna IAM.

Konteks enkripsi dalam kebijakan

Konteks enkripsi digunakan terutama untuk memverifikasi integritas dan keaslian. Tetapi Anda juga dapat menggunakan konteks enkripsi untuk mengontrol akses ke enkripsi simetris AWS KMS keys dalam kebijakan utama dan kebijakan IAM.

Kunci EncryptionContextKeys kondisi kms:EncryptionContext: dan kms: mengizinkan (atau menolak) izin hanya jika permintaan menyertakan kunci konteks enkripsi tertentu atau pasangan kunci-nilai.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan `RoleForExampleApp` peran untuk menggunakan kunci KMS dalam `Decrypt` operasi. Ini menggunakan kunci `kms:EncryptionContext:`*`context-key`* kondisi untuk mengizinkan izin ini hanya ketika konteks enkripsi dalam permintaan menyertakan pasangan konteks `AppName:ExampleApp` enkripsi.

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

```
    }
  }
}
```

Untuk informasi lebih lanjut tentang kunci kondisi konteks enkripsi, lihat Kunci kondisi untuk AWS KMS.

Konteks enkripsi dalam hibah

Saat Anda membuat hibah, Anda dapat menyertakan batasan hibah yang menetapkan kondisi untuk izin hibah. AWS KMS mendukung dua batasan hibah, `EncryptionContextEquals` dan `EncryptionContextSubset`, yang keduanya melibatkan konteks enkripsi dalam permintaan untuk operasi kriptografi. Ketika Anda menggunakan batasan hibah ini, izin dalam hibah hanya efektif ketika konteks enkripsi dalam permintaan untuk operasi kriptografi memenuhi persyaratan batasan hibah.

Misalnya, Anda dapat menambahkan batasan `EncryptionContextEquals` hibah ke hibah yang memungkinkan operasi. GenerateDataKey Dengan batasan ini, hibah memungkinkan operasi hanya ketika konteks enkripsi dalam permintaan cocok huruf besar/kecil untuk konteks enkripsi dalam batasan hibah.

```
$ 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}
```

Permintaan seperti berikut dari prinsipal penerima hibah akan memenuhi batasan `EncryptionContextEquals`.

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

Untuk detail tentang batasan hibah, lihat Menggunakan batas pemberian izin. Untuk informasi mendetail tentang hibah, lihat the section called “Izin”.

Konteks enkripsi pencatatan

AWS KMSdigunakan AWS CloudTrail untuk mencatat konteks enkripsi sehingga Anda dapat menentukan kunci dan data KMS mana yang telah diakses. Entri log menunjukkan dengan tepat kunci KMS mana yang digunakan untuk mengenkripsi atau mendekripsi data tertentu yang direferensikan oleh konteks enkripsi dalam entri log.

> **Important**
>
> Karena konteks enkripsi dicatat, konteks tidak boleh berisi informasi sensitif.

Menyimpan konteks enkripsi

Untuk menyederhanakan penggunaan konteks enkripsi ketika Anda memanggil operasi `Decrypt` atau `ReEncrypt`, Anda dapat menyimpan konteks enkripsi bersama data terenkripsi. Sebaiknya Anda hanya menyimpan konteks enkripsi yang mencukupi untuk membantu Anda membuat konteks enkripsi penuh ketika Anda membutuhkannya untuk enkripsi atau dekripsi.

Misalnya, jika konteks enkripsi sepenuhnya memenuhi syarat untuk file, simpan hanya bagian dari jalur tersebut dengan konten file terenkripsi. Kemudian, ketika Anda membutuhkan konteks enkripsi penuh, rekonstruksi konteks dari fragmen yang disimpan. Jika seseorang merusak file, seperti mengganti namanya atau memindahkannya ke lokasi lain, perubahan nilai konteks enkripsi dan permintaan dekripsi akan gagal.

Kebijakan kunci

Saat Anda membuat kunci KMS, Anda menentukan siapa yang dapat menggunakan dan mengelola kunci KMS itu. Izin ini disertakan dalam dokumen yang disebut kebijakan kunci. Anda dapat menggunakan kebijakan kunci untuk menambah, menghapus, atau mengubah izin kapan saja untuk kunci yang dikelola pelanggan. Tetapi Anda tidak dapat mengedit kebijakan kunci untuk fileKunci yang dikelola AWS. Untuk informasi selengkapnya, lihat Kebijakan utama di AWS KMS.

Pemberian Izin

Hibah adalah instrumen kebijakan yang memungkinkan AWS prinsipal untuk digunakan AWS KMS keys dalam operasi kriptografi. Hal ini juga dapat membiarkan mereka melihat kunci KMS (DescribeKey) dan membuat dan mengelola hibah. Saat mengotorisasi akses ke kunci KMS, hibah dipertimbangkan bersama dengan kebijakan utama dan kebijakan IAM. Hibah sering digunakan

untuk izin sementara karena Anda dapat membuatnya, menggunakan izinnya, dan menghapusnya tanpa mengubah kebijakan kunci atau kebijakan IAM. Karena hibah bisa sangat spesifik, serta mudah dibuat dan dicabut, hibah tersebut sering digunakan untuk memberikan izin sementara atau izin yang lebih terperinci.

Untuk informasi mendetail tentang hibah, termasuk terminologi hibah, lihat Hibah di AWS KMS.

Mengaudit penggunaan kunci KMS

Anda dapat menggunakan AWS CloudTrail untuk mengaudit penggunaan kunci. CloudTrail membuat file log yang berisi riwayat panggilan AWS API dan peristiwa terkait untuk akun Anda. File log ini mencakup semua permintaan AWS KMS API yang dibuat dengan Konsol Manajemen AWS, AWS SDK, dan alat baris perintah. File log juga mencakup permintaan untuk AWS KMS yang dibuat oleh layanan AWS atas nama Anda. Anda dapat menggunakan file log ini untuk menemukan informasi penting, termasuk kapan kunci KMS digunakan, operasi yang diminta, identitas pemohon, dan alamat IP sumber. Untuk informasi selengkapnya, lihat Logging dengan AWS CloudTrail dan Panduan Pengguna AWS CloudTrail.

Infrastruktur manajemen kunci

Praktik umum dalam kriptografi adalah mengenkripsi dan mendekripsi dengan algoritme yang tersedia untuk umum dan peer-review seperti AES (Advanced Encryption Standard) dan kunci rahasia. Salah satu masalah utama dengan kriptografi adalah ia sangat sulit untuk menyimpan rahasia kunci. Ini biasanya pekerjaan infrastruktur manajemen kunci (KMI). AWS KMSmengoperasikan infrastruktur utama untuk Anda. AWS KMSmembuat dan menyimpan kunci root Anda dengan aman, disebut AWS KMS keys. Untuk informasi lebih lanjut tentang bagaimana AWS KMS beroperasi, lihat Detail Kriptografi AWS Key Management Service.

Mengelola kunci

Untuk memulaiAWS KMS, buat file AWS KMS key.

Topik di bagian ini menjelaskan cara mengelola kunci KMS dasar, kunci KMS enkripsi simetris, dari pembuatan hingga penghapusan. Ini mencakup topik tentang mengedit dan melihat tombol, menandai tombol, mengaktifkan dan menonaktifkan tombol, memutar materi kunci, dan menggunakan AWS alat dan layanan untuk memantau penggunaan tombol KMS Anda. Ini juga mencakup informasi tentang penggunaan AWS CloudFormation untuk membuat dan mengelola kunci KMS Anda dan referensi status kunci yang menunjukkan status kunci yang diperlukan untuk setiap AWS KMS operasi.

Untuk informasi tentang membuat, menggunakan, dan mengelola jenis kunci KMS lainnya, lihatKunci tujuan khusus.

Topik

Membuat kunci

Anda dapat membuat AWS KMS keys diAWS Management Console, atau dengan menggunakan CreateKeyoperasi atau AWS CloudFormationtemplate. Selama proses ini, Anda memilih jenis kunci KMS, regionalitasnya (Single-region atau Multi-region), dan asal bahan kunci (secara default, AWS

KMS menciptakan materi kunci). Anda tidak dapat mengubah properti ini setelah kunci KMS dibuat. Anda juga menetapkan kebijakan kunci untuk kunci KMS, yang dapat Anda ubah kapan saja.

Topik ini menjelaskan cara membuat kunci KMS dasar, kunci KMS enkripsi simetris untuk satu Wilayah dengan materi kunci dari. AWS KMS Anda dapat menggunakan kunci KMS ini untuk melindungi sumber daya Anda di fileLayanan AWS. Untuk informasi rinci tentang kunci KMS enkripsi simetris, lihat. Spesifikasi kunci SYMMETRIC_DEFAULT Untuk bantuan membuat jenis kunci lainnya, lihatKunci tujuan khusus.

Jika Anda membuat kunci KMS untuk mengenkripsi data yang Anda simpan atau kelola dalam suatu AWS layanan, buat kunci KMS enkripsi simetris. AWSlayanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris. Untuk bantuan menentukan jenis kunci KMS yang akan dibuat, lihatMemilih tipe kunci KMS.

> **Note**
>
> Kunci KMS simetris sekarang disebut kunci KMS enkripsi simetris. AWS KMSmendukung dua jenis kunci KMS simetris, kunci KMS enkripsi simetris (tipe default) dan kunci HMAC KMS, yang juga merupakan kunci simetris.

Ketika Anda membuat kunci KMS di AWS KMS konsol, Anda diminta untuk memberikan alias (nama ramah). `CreateKey`Operasi tidak membuat alias untuk kunci KMS baru. Untuk membuat alias untuk kunci KMS baru atau yang sudah ada, gunakan operasi. CreateAlias Untuk informasi mendetail tentang alias di AWS KMS, lihat Menggunakan alias.

Topik ini menjelaskan cara membuat kunci KMS enkripsi simetris. Gunakan tabel berikut untuk menemukan petunjuk untuk membuat kunci KMS dari berbagai jenis.

Petunjuk untuk membuat kunci KMS

Jenis kunci KMS	Petunjuk
Kunci enkripsi simetris (SYMMETRIC_DEFAULT)	the section called “Membuat kunci KMS enkripsi simetris”
Kunci asimetris	the section called “Membuat tombol KMS asimetris”

Jenis kunci KMS	Petunjuk
Kunci HMAC	the section called “Membuat kunci HMAC”
Kunci Multi-Region (dari jenis apa pun)	the section called “Membuat kunci primer dengan materi kunci yang diimpor” the section called “Membuat kunci replika dengan materi kunci yang diimpor”
Bahan kunci impor (“Bawa kunci Anda sendiri - BYOK”)	the section called “Langkah 1: Buat kunci KMS tanpa bahan kunci”
AWS CloudHSMtoko kunci	the section called “Membuat kunci KMS di toko AWS CloudHSM kunci”
Toko kunci eksternal (“Tahan kunci Anda sendiri - HYOK”)	the section called “Membuat kunci KMS di toko kunci eksternal”

Pelajari lebih lanjut:

- Untuk membuat kunci data untuk enkripsi sisi klien, gunakan operasi. GenerateDataKey
- Untuk membuat kunci KMS asimetris untuk enkripsi atau penandatanganan, lihat. Membuat tombol KMS asimetris
- Untuk membuat kunci HMAC KMS, lihat. Membuat kunci HMAC KMS
- Untuk membuat kunci KMS dengan materi kunci impor (“bawa kunci Anda sendiri”), lihatMengimpor bahan kunci langkah 1: Buat materi AWS KMS key tanpa kunci.
- Untuk membuat kunci primer Multi-wilayah atau kunci replika, lihat. Membuat kunci multi-Wilayah
- Untuk membuat kunci KMS di toko kunci kustom (asal materi utama adalah Custom Key Store (CloudHSM)), lihat. Membuat kunci KMS di toko AWS CloudHSM kunci
- Untuk menggunakan AWS CloudFormation template untuk membuat kunci KMS, lihat AWS::KMS::Keydi Panduan AWS CloudFormation Pengguna.
- Untuk menentukan apakah kunci KMS yang ada simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

- Untuk menggunakan kunci KMS Anda secara terprogram dan dalam operasi antarmuka baris perintah, Anda memerlukan ID kunci atau kunci ARN. Untuk petunjuk mendetail, lihat Menemukan ID kunci dan kunci ARN.
- Untuk informasi tentang kuota yang berlaku untuk kunci KMS, lihat. Kuota

Topik

- Izin untuk membuat kunci KMS
- Membuat kunci KMS enkripsi simetris

Izin untuk membuat kunci KMS

Untuk membuat kunci KMS di konsol atau dengan menggunakan API, Anda harus memiliki izin berikut dalam kebijakan IAM. Sebisa mungkin, gunakan kunci kondisi untuk membatasi izin. Misalnya, Anda dapat menggunakan kunci KeySpec kondisi kms: dalam kebijakan IAM untuk mengizinkan prinsipal membuat hanya kunci enkripsi simetris.

Untuk contoh kebijakan IAM untuk prinsipal yang membuat kunci, lihat Izinkan pengguna untuk membuat kunci KMS.

> **Note**
>
> Berhati-hatilah saat memberikan izin prinsipal untuk mengelola tanda dan alias. Mengubah tag atau alias dapat mengizinkan atau menolak izin ke kunci yang dikelola pelanggan. Untuk detailnya, lihat ABAC untuk AWS KMS.

- kms: CreateKey diperlukan.
- kms: CreateAlias diperlukan untuk membuat kunci KMS di konsol di mana alias diperlukan untuk setiap kunci KMS baru.
- kms: TagResource diperlukan untuk menambahkan tag saat membuat kunci KMS.
- iam: CreateServiceLinkedRole diperlukan untuk membuat kunci utama Multi-wilayah. Untuk detailnya, lihat Mengontrol akses ke kunci multi-Wilayah.

PutKeyPolicyIzin kms: tidak diperlukan untuk membuat kunci KMS. Izin `kms:CreateKey` termasuk izin untuk menetapkan kebijakan kunci awal. Tetapi Anda harus menambahkan izin ini ke kebijakan

kunci saat membuat kunci KMS untuk memastikan bahwa Anda dapat mengontrol akses ke kunci KMS. Alternatifnya adalah menggunakan BypassLockoutSafetyCheckparameter, yang tidak disarankan.

Kunci KMS milik AWS akun tempat mereka dibuat. Pengguna IAM yang membuat kunci KMS tidak dianggap sebagai pemilik kunci dan mereka tidak secara otomatis memiliki izin untuk menggunakan atau mengelola kunci KMS yang mereka buat. Seperti prinsipal lainnya, pembuat kunci perlu mendapatkan izin melalui kebijakan kunci, kebijakan IAM, atau hibah. Namun, kepala sekolah yang memiliki `kms:CreateKey` izin dapat mengatur kebijakan kunci awal dan memberi diri mereka izin untuk menggunakan atau mengelola kunci tersebut.

Membuat kunci KMS enkripsi simetris

Anda dapat membuat kunci KMS di AWS Management Console atau dengan menggunakan AWS KMS API.

Topik ini menjelaskan cara membuat kunci KMS dasar, kunci KMS enkripsi simetris untuk satu Wilayah dengan materi kunci dari. AWS KMS Anda dapat menggunakan kunci KMS ini untuk melindungi sumber daya Anda di fileLayanan AWS. Untuk bantuan membuat jenis kunci lainnya, lihatKunci tujuan khusus.

Membuat kunci KMS enkripsi simetris (konsol)

Anda dapat menggunakan AWS Management Console untuk membuat AWS KMS keys (kunci KMS).

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Untuk membuat kunci KMS enkripsi simetris, untuk Key type pilih Symmetric.

Untuk informasi tentang cara membuat kunci KMS asimetris di AWS KMS konsol, lihat. Membuat tombol KMS asimetris (konsol)

6. Dalam Penggunaan kunci, opsi Enkripsi dan dekripsi dipilih untuk Anda.

   Untuk informasi tentang cara membuat kunci KMS yang menghasilkan dan memverifikasi kode MAC, lihatMembuat kunci HMAC KMS.

7. Pilih Berikutnya.

   Untuk informasi tentang opsi lanjutan, lihatKunci tujuan khusus.

8. Ketik alias untuk kunci KMS. Nama alias tidak dapat dimulai dengan **`aws/`**. **`aws/`**Awalan dicadangkan oleh Amazon Web Services untuk mewakili Kunci yang dikelola AWS di akun Anda.

   > **Note**
   >
   > Menambahkan, menghapus, atau memperbarui alias dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan alias untuk mengontrol akses ke tombol KMS.

   Alias adalah nama tampilan yang dapat Anda gunakan untuk mengidentifikasi kunci KMS. Kami menyarankan Anda memilih alias yang menunjukkan jenis data yang Anda rencanakan untuk dilindungi atau aplikasi yang Anda rencanakan untuk digunakan dengan kunci KMS.

   Alias diperlukan saat Anda membuat kunci KMS di file. AWS Management Console Mereka opsional saat Anda menggunakan CreateKeyoperasi.

9. (Opsional) Ketik deskripsi untuk kunci KMS.

   Anda dapat menambahkan deskripsi sekarang atau memperbaruinya kapan saja kecuali status kuncinya adalah `Pending Deletion` atau`Pending Replica Deletion`. Untuk menambah, mengubah, atau menghapus deskripsi kunci terkelola pelanggan yang ada, edit deskripsi di AWS Management Console atau gunakan UpdateKeyDescriptionoperasi.

10. (Opsional) Ketik kunci tanda dan nilai tanda opsional. Untuk menambahkan lebih dari satu tag ke tombol KMS, pilih Tambah tag.

> **Note**
>
> Menandai atau melepas tag kunci KMS dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.

Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda Anda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS

11. Pilih Berikutnya.
12. Pilih pengguna IAM dan peran yang dapat mengelola kunci KMS.

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini.
> Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk memberikan izin kepada prinsipal lain untuk mengelola kunci KMS. Untuk detailnya, lihat the section called “Kebijakan kunci default”.
>
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

13. (Opsional) Untuk mencegah pengguna dan peran IAM yang dipilih menghapus kunci KMS ini, di bagian Penghapusan kunci di bagian bawah halaman, kosongkan kotak centang Izinkan administrator kunci untuk menghapus kunci ini.
14. Pilih Berikutnya.
15. Pilih pengguna IAM dan peran yang dapat menggunakan kunci dalam operasi kriptografi

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini.
> Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk

memberikan izin kepada prinsipal lain untuk menggunakan kunci KMS dalam operasi kriptografi. Untuk detailnya, lihat the section called “Kebijakan kunci default”.

Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

16. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, dalam bagian Lainnya Akun AWS di bawah halaman, pilih Tambahkan Akun AWS lain dan masukkan nomor identifikasi Akun AWS akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Untuk mengizinkan prinsipal di akun eksternal menggunakan kunci KMS, Administrator akun eksternal harus membuat kebijakan IAM yang memberikan izin ini. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

17. Pilih Selanjutnya.
18. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
19. Pilih Selesai untuk membuat kunci KMS.

Membuat kunci KMS enkripsi simetris (API) AWS KMS

Anda dapat menggunakan CreateKeyoperasi untuk membuat semua AWS KMS keys jenis. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Operasi berikut menciptakan kunci KMS yang paling umum digunakan, kunci enkripsi simetris dalam satu Wilayah yang didukung oleh materi kunci yang dihasilkan oleh. AWS KMS Operasi ini tidak memiliki parameter yang diperlukan. Namun, Anda mungkin juga ingin menggunakan parameter `Policy` untuk menentukan kebijakan kunci. Anda dapat mengubah kebijakan kunci (PutKeyPolicy) dan menambahkan elemen opsional, seperti deskripsi dan tag kapan saja. Anda juga dapat membuat kunci asimetris, kunci multi-wilayah, kunci dengan bahan kunci yang diimpor, dan kunci di toko kunci khusus.

`CreateKey`Operasi tidak memungkinkan Anda menentukan alias, tetapi Anda dapat menggunakan CreateAliasoperasi untuk membuat alias untuk kunci KMS baru Anda.

Berikut ini adalah contoh panggilan ke operasi `CreateKey` tanpa parameter. Perintah ini menggunakan semua nilai default. Ini menciptakan kunci KMS enkripsi simetris dengan bahan kunci yang dihasilkan oleh. AWS KMS

```
$ aws kms create-key
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "MultiRegion": false
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
    }
}
```

Jika Anda tidak menentukan kebijakan kunci untuk kunci KMS baru, kebijakan kunci default yang `CreateKey` berlaku berbeda dari kebijakan kunci default yang diterapkan konsol saat Anda menggunakannya untuk membuat kunci KMS baru.

Misalnya, panggilan ke GetKeyPolicyoperasi ini mengembalikan kebijakan kunci yang `CreateKey` berlaku. Ini memberikan Akun AWS akses ke kunci KMS dan memungkinkannya untuk membuat kebijakan AWS Identity and Access Management (IAM) untuk kunci KMS. Untuk informasi rinci tentang kebijakan IAM dan kebijakan utama untuk kunci KMS, lihat Kontrol autentikasi dan akses untuk 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" : "*"
  } ]
}
```

Menggunakan alias

Alias adalah nama yang ramah untuk a AWS KMS key. Misalnya, alias memungkinkan Anda merujuk ke kunci KMS sebagai `test-key` gantinya. `1234abcd-12ab-34cd-56ef-1234567890ab`

Anda dapat menggunakan alias untuk mengidentifikasi kunci KMS di AWS KMS konsol, dalam operasi, dan dalam DescribeKeyoperasi kriptografi, seperti Enkripsi dan. GenerateDataKey Alias juga memudahkan untuk mengenali file Kunci yang dikelola AWS. Alias untuk kunci KMS ini selalu memiliki formulir. `aws/<service-name>` Misalnya, alias Kunci yang dikelola AWS untuk Amazon DynamoDB adalah. `aws/dynamodb` Anda dapat menetapkan standar alias yang serupa untuk proyek Anda, seperti mendahului nama alias Anda dengan nama proyek atau kategori.

Anda juga dapat mengizinkan dan menolak akses ke kunci KMS berdasarkan aliasnya tanpa mengedit kebijakan atau mengelola hibah. Fitur ini adalah bagian dari dukungan AWS KMS untuk kontrol akses berbasis atribut (ABAC). Untuk rincian selengkapnya, lihat Menggunakan alias untuk mengontrol akses ke tombol KMS.

Sebagian besar kekuatan alias berasal dari kemampuan Anda untuk mengubah kunci KMS yang terkait dengan alias kapan saja. Alias dapat membuat kode Anda lebih mudah ditulis dan dipelihara. Misalnya, Anda menggunakan alias untuk merujuk ke kunci KMS tertentu dan Anda ingin mengubah kunci KMS. Dalam hal ini, cukup kaitkan alias dengan kunci KMS yang berbeda. Anda tidak perlu mengubah kode.

Alias juga mempermudah menggunakan kembali kode yang sama di Wilayah AWS berbeda. Buat alias dengan nama yang sama di beberapa Wilayah dan kaitkan setiap alias dengan kunci KMS di Wilayahnya. Ketika kode berjalan di setiap Wilayah, alias mengacu pada kunci KMS terkait di Wilayah tersebut. Lihat contohnya di Menggunakan alias dalam aplikasi Anda.

Anda dapat membuat alias untuk kunci KMS di AWS KMS konsol, dengan menggunakan CreateAliasAPI, atau dengan menggunakan template. AWS CloudFormation

API AWS KMS menyediakan kontrol penuh atas alias di setiap akun dan Wilayah. API mencakup operasi untuk membuat alias (CreateAlias), melihat nama alias dan alias ARN (ListAliases), mengubah kunci KMS yang terkait dengan alias (UpdateAlias), dan menghapus alias (). DeleteAlias Untuk contoh mengelola alias beberapa bahasa pemrograman, lihat the section called “Bekerja dengan alias”.

Sumber daya berikut dapat membantu Anda mempelajari selengkapnya:

- Untuk informasi tentang pengidentifikasi kunci KMS, termasuk alias, lihat. Pengidentifikasi kunci () KeyId
- Untuk bantuan menggunakan AWS CloudFormation template untuk membuat alias untuk kunci KMS, lihat AWS::KMS::Aliasdi AWS CloudFormationPanduan Pengguna.
- Untuk bantuan menemukan alias yang terkait dengan kunci KMS, lihat Menemukan nama alias dan ARN alias
- Untuk informasi tentang kuota sumber daya untuk alias dan kuota tarif untuk operasi API yang terkait dengan alias, lihat Kuota.
- Untuk contoh membuat dan mengelola alias dalam beberapa bahasa pemrograman, lihat Bekerja dengan alias.

Topik

- Tentang alias
- Mengelola alias
- Menggunakan alias dalam aplikasi Anda

- Mengontrol akses ke alias
- Menggunakan alias untuk mengontrol akses ke tombol KMS
- Menemukan alias dalam log AWS CloudTrail

Tentang alias

Pelajari cara kerja alias di AWS KMS.

Alias adalah sumber daya AWS independen

Alias bukan milik kunci KMS. Tindakan yang Anda lakukan pada alias tidak memengaruhi kunci KMS yang terkait. Anda dapat membuat alias untuk kunci KMS dan kemudian memperbarui alias sehingga dikaitkan dengan kunci KMS yang berbeda. Anda bahkan dapat menghapus alias tanpa efek apa pun pada kunci KMS terkait. Namun, jika Anda menghapus kunci KMS, semua alias yang terkait dengan kunci KMS tersebut akan dihapus.

Jika Anda menetapkan alias sebagai sumber daya dalam kebijakan IAM, kebijakan tersebut merujuk ke alias, bukan ke kunci KMS terkait.

Setiap alias memiliki dua format

Saat membuat alias, Anda menentukan nama alias. AWS KMS membuat ARN alias untuk Anda.

- ARN alias adalah Amazon Resource Name (ARN) yang secara unik mengidentifikasi alias.

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

- Nama alias yang unik di akun dan Wilayah. Dalam API AWS KMS, nama alias selalu diberi prefiks dengan `alias/`. Prefiks itu dihilangkan dalam konsol AWS KMS.

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

Alias bukan rahasia

Alias dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya. Jangan sertakan informasi rahasia atau sensitif dalam nama alias.

Setiap alias dikaitkan dengan satu kunci KMS pada satu waktu

Alias dan kunci KMS-nya harus berada di akun dan Wilayah yang sama.

Anda dapat mengaitkan alias dengan kunci yang dikelola pelanggan apa pun di area yang sama Akun AWS dan Wilayah. Namun, Anda tidak memiliki izin untuk mengaitkan alias dengan Kunci yang dikelola AWS.

Misalnya, ListAliasesoutput ini menunjukkan bahwa `test-key` alias dikaitkan dengan tepat satu kunci KMS target, yang diwakili oleh properti. `TargetKeyId`

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
}
```

Beberapa alias dapat dikaitkan dengan kunci KMS yang sama

Misalnya, Anda dapat mengaitkan `test-key` dan `project-key` alias dengan kunci KMS yang sama.

```
{
     "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
}
```

Alias harus unik di akun dan Wilayah

Misalnya, Anda hanya dapat memiliki satu alias `test-key` di setiap akun dan Wilayah. Alias peka huruf besar/kecil, tetapi alias yang berbeda hanya dalam kapitalisasi mereka sangat rentan terhadap kesalahan. Anda tidak dapat mengubah nama alias. Namun, Anda dapat menghapus alias dan membuat alias baru dengan nama yang diinginkan.

Anda dapat membuat alias dengan nama yang sama di Wilayah yang berbeda

Misalnya, Anda dapat memiliki alias `finance-key` di AS Timur (Virginia U.) dan alias `finance-key` di Eropa (Frankfurt). Setiap alias akan dikaitkan dengan kunci KMS di Wilayahnya. Jika kode Anda merujuk pada nama alias seperti `alias/finance-key`, Anda dapat menjalankannya di beberapa Wilayah. Di setiap Wilayah, ia menggunakan kunci KMS yang berbeda. Untuk detailnya, lihat Menggunakan alias dalam aplikasi Anda.

Anda dapat mengubah kunci KMS yang terkait dengan alias

Anda dapat menggunakan UpdateAliasoperasi untuk mengaitkan alias dengan kunci KMS yang berbeda. Misalnya, jika `finance-key` alias dikaitkan dengan kunci `1234abcd-12ab-34cd-56ef-1234567890ab` KMS, Anda dapat memperbaruinya sehingga dikaitkan dengan kunci `0987dcba-09fe-87dc-65ba-ab0987654321` KMS.

Namun, kunci KMS saat ini dan yang baru harus memiliki tipe yang sama (keduanya simetris atau keduanya asimetris atau keduanya HMAC), dan mereka harus memiliki penggunaan kunci yang sama (ENCRYPT_DECRYPT atau SIGN_VERIFY atau GENERATE_VERIFY_MAC). Pembatasan ini mencegah kesalahan dalam kode yang menggunakan alias. Jika Anda harus mengaitkan alias dengan jenis kunci yang berbeda, dan Anda telah mengurangi risiko, Anda dapat menghapus dan membuat ulang alias.

Beberapa kunci KMS tidak memiliki alias

Ketika Anda membuat kunci KMS di AWS KMS konsol, Anda harus memberikan alias baru. Tetapi alias tidak diperlukan saat Anda menggunakan CreateKeyoperasi untuk membuat kunci KMS. Selain itu, Anda dapat menggunakan UpdateAliasoperasi untuk mengubah kunci KMS yang terkait dengan alias dan DeleteAliasoperasi untuk menghapus alias. Akibatnya, beberapa kunci KMS mungkin memiliki beberapa alias, dan beberapa mungkin tidak memilikinya.

AWS membuat alias di akun Anda

AWSmembuat alias di akun Anda untuk Kunci yang dikelola AWS. Alias ini memiliki nama formulir `alias/aws/<service-name>`, seperti `alias/aws/s3`.

Beberapa AWS alias tidak memiliki kunci KMS. Alias yang telah ditentukan ini biasanya dikaitkan dengan Kunci yang dikelola AWS ketika Anda mulai menggunakan layanan.

Gunakan alias untuk mengidentifikasi kunci KMS

Anda dapat menggunakan nama alias atau alias ARN untuk mengidentifikasi kunci KMS dalam operasi kriptografi,, dan. DescribeKeyGetPublicKey (Jika kunci KMS berbeda Akun AWS, Anda

harus menggunakan kunci ARN atau alias ARN.) Alias bukan pengidentifikasi yang valid untuk kunci KMS dalam operasi lain. AWS KMS Untuk informasi tentang pengidentifikasi kunci yang valid untuk setiap operasi API AWS KMS, lihat deskripsi parameter `KeyId` dalam Referensi API AWS Key Management Service.

Anda tidak dapat menggunakan nama alias atau alias ARN untuk mengidentifikasi kunci KMS dalam kebijakan IAM. Untuk mengontrol akses ke kunci KMS berdasarkan aliasnya, gunakan kunci kondisi kms: RequestAlias atau kms:. ResourceAliases Untuk rincian selengkapnya, lihat ABAC untuk AWS KMS.

Mengelola alias

Pengguna yang diotorisasi dapat membuat, melihat, dan menghapus alias. Anda juga dapat memperbarui alias, yaitu mengaitkan alias yang ada dengan kunci KMS yang berbeda.

Topik

- Membuat alias
- Melihat alias
- Memperbarui alias
- Menghapus alias

Membuat alias

Anda dapat membuat alias di konsol AWS KMS atau dengan menggunakan operasi API AWS KMS.

Alias harus string berisi 1-256 karakter. Ini hanya dapat berisi karakter alfanumerik, garis miring depan (/), garis bawah (_), dan tanda garis (-). Nama alias untuk kunci yang dikelola pelanggan tidak dapat dimulai dengan`alias/aws/`. `alias/aws/`Awalan dicadangkan untuk Kunci yang dikelola AWS.

Anda dapat membuat alias untuk kunci KMS baru atau untuk kunci KMS yang ada. Anda dapat menambahkan alias sehingga kunci KMS tertentu digunakan dalam proyek atau aplikasi.

Buat alias (konsol)

Saat Anda membuat kunci KMS di AWS KMS konsol, Anda harus membuat alias untuk kunci KMS baru. Untuk membuat alias untuk kunci KMS yang ada, gunakan tab Alias pada halaman detail untuk kunci KMS.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. Anda tidak dapat mengelola alias untuk Kunci yang dikelola AWS atauKunci milik AWS.
4. Dalam tabel, pilih ID kunci atau alias kunci KMS. Kemudian, pada halaman detail kunci KMS, pilih tab Alias.

   Jika kunci KMS memiliki beberapa alias, kolom Alias dalam tabel menampilkan satu alias dan ringkasan alias, seperti (+ n lebih). Memilih ringkasan alias akan membawa Anda langsung ke tab Alias pada halaman detail kunci KMS.
5. Pada tab Alias, pilih Buat alias. Masukkan nama alias dan pilih Buat alias.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

> **Note**
>
> Jangan tambahkan `alias/` awalan. Konsol secara otomatis menambahkannya untuk Anda. Jika Anda memasukkan `alias/ExampleAlias`, nama alias sebenarnya akan `alias/alias/ExampleAlias`.

Buat alias (API AWS KMS)

Untuk membuat alias, gunakan CreateAliasoperasi. Berbeda dengan proses pembuatan kunci KMS di konsol, CreateKeyoperasi tidak membuat alias untuk kunci KMS baru.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

Anda dapat menggunakan `CreateAlias` operasi untuk membuat alias untuk kunci KMS baru tanpa alias. Anda juga dapat menggunakan `CreateAlias` operasi untuk menambahkan alias ke kunci KMS yang ada atau untuk membuat ulang alias yang secara tidak sengaja dihapus.

Di operasi API AWS KMS, nama alias harus dimulai dengan `alias/` diikuti dengan nama, misalnya `alias/ExampleAlias`. Alias harus unik di akun dan Wilayah. Untuk menemukan nama alias yang sudah digunakan, gunakan ListAliasesoperasi. Nama alias peka terhadap huruf besar-kecil.

`TargetKeyId`Dapat berupa kunci yang dikelola pelanggan dalam hal yang samaWilayah AWS. Untuk mengidentifikasi kunci KMS, gunakan ID kunci atau kunci ARN. Anda tidak dapat menggunakan alias lain.

Contoh berikut membuat `example-key` alias dan mengaitkannya dengan kunci KMS tertentu. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI). Untuk contoh dalam beberapa bahasa pemrograman, lihat Bekerja dengan alias.

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

`CreateAlias`tidak mengembalikan output apa pun. Untuk melihat alias baru, gunakan operasi `ListAliases`. Untuk rincian selengkapnya, lihat Melihat alias (API AWS KMS).

Melihat alias

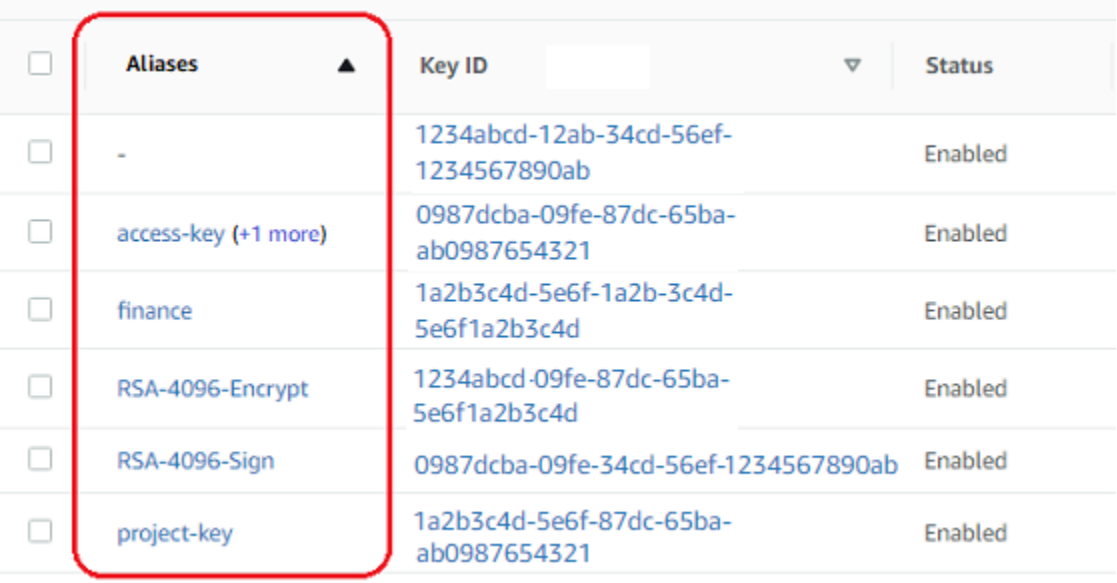
Alias memudahkan untuk mengenali kunci KMS di konsol. AWS KMS Anda dapat melihat alias untuk kunci KMS di AWS KMS konsol atau dengan menggunakan operasi. ListAliases DescribeKeyOperasi, yang mengembalikan properti kunci KMS, tidak termasuk alias.

Melihat alias (konsol)

Kunci dan Kunci yang dikelola AWShalaman yang dikelola Pelanggan di AWS KMS konsol menampilkan alias yang terkait dengan setiap kunci KMS. Anda juga dapat mencari, mengurutkan, dan memfilter kunci KMS berdasarkan aliasnya.

Gambar konsol AWS KMS berikut menunjukkan alias pada halaman Kunci terkelola pelanggan dari akun contoh. Seperti yang ditunjukkan pada gambar, beberapa tombol KMS tidak memiliki alias.

Ketika kunci KMS memiliki beberapa alias, kolom Alias menampilkan satu alias dan ringkasan alias (+ n lebih). Ringkasan alias menunjukkan berapa banyak alias tambahan yang terkait dengan kunci KMS dan tautan ke tampilan semua alias untuk kunci KMS pada tab Alias.

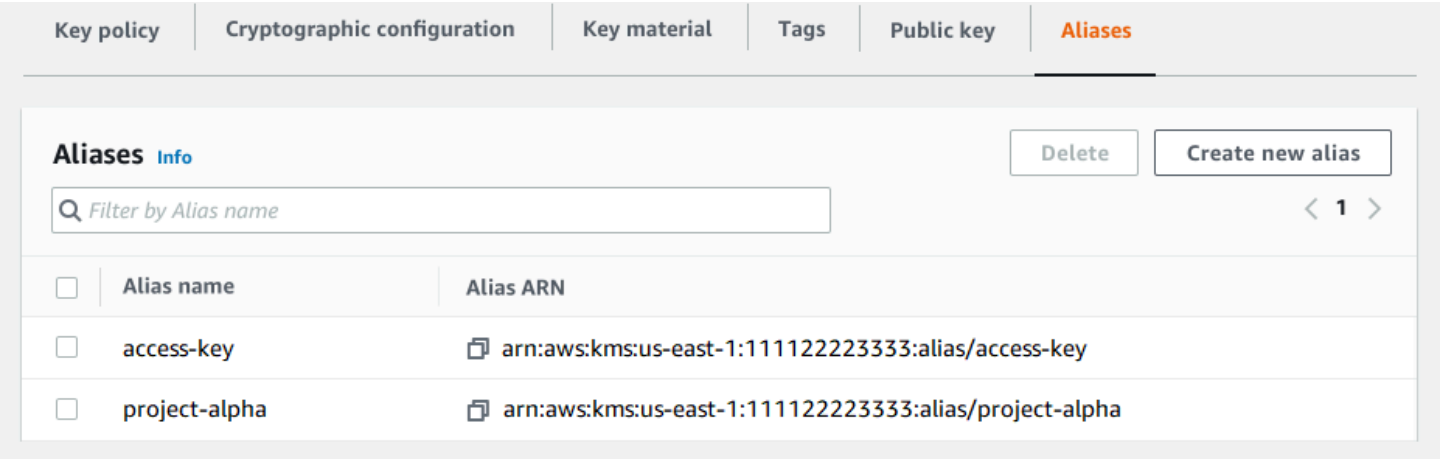


Tab Alias pada halaman detail untuk setiap tombol KMS menampilkan nama alias dan alias ARN dari semua alias untuk kunci KMS di dan Wilayah. Akun AWS Anda juga dapat menggunakan tab Alias untuk membuat alias dan menghapus alias.

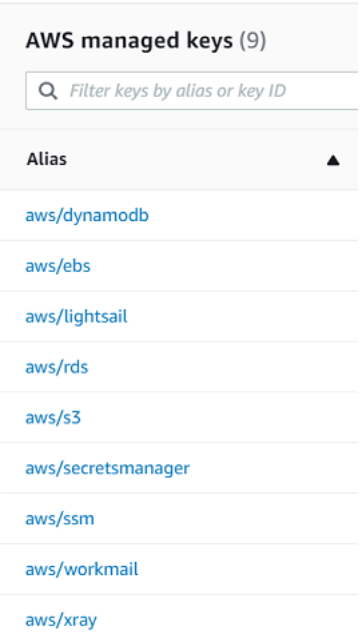
Untuk menemukan nama alias dan alias ARN dari semua alias untuk kunci KMS, gunakan tab Alias.

- Untuk pergi langsung ke tab Alias, di kolom Alias, pilih ringkasan alias (+n lebih banyak). Ringkasan alias hanya muncul jika kunci KMS memiliki lebih dari satu alias.
- Atau, pilih alias atau ID kunci dari kunci KMS (yang membuka halaman detail untuk tombol KMS) dan kemudian pilih tab Alias. Tab ada di bawah bagian Konfigurasi umum.

Gambar berikut menunjukkan tab Alias untuk kunci KMS contoh.



Anda dapat menggunakan alias untuk mengenaliKunci yang dikelola AWS, seperti yang ditunjukkan pada Kunci yang dikelola AWShalaman contoh ini. Alias untuk Kunci yang dikelola AWS selalu memiliki format:`aws/<service-name>`. Misalnya, alias Kunci yang dikelola AWS untuk Amazon DynamoDB adalah. `aws/dynamodb`



Melihat alias (API AWS KMS)

ListAliasesOperasi mengembalikan nama alias dan alias ARN alias di akun dan Wilayah. Outputnya mencakup alias untuk Kunci yang dikelola AWS dan untuk kunci yang dikelola pelanggan. Alias untuk Kunci yang dikelola AWS memiliki formatAWS/`<service-name>`, sepertiaws/dynamodb.

Respons mungkin juga menyertakan alias yang tidak memiliki bidang `TargetKeyId`. Ini adalah alias yang AWS telah ditentukan sebelumnya yang telah dibuat tetapi belum dikaitkan dengan kunci KMS.

```
$ aws kms list-aliases
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
```

```
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        }
    ]
}
```


Untuk mendapatkan semua alias yang terkait dengan kunci KMS tertentu, gunakan `KeyId` parameter opsional operasi. `ListAliases` `KeyId`Parameter mengambil ID kunci atau kunci ARN dari kunci KMS.

Contoh ini mendapatkan semua alias yang terkait dengan kunci `0987dcba-09fe-87dc-65ba-ab0987654321` KMS.

```
$ 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
        }
    ]
}
```

Parameter `KeyId` tidak mengambil karakter wildcard, tetapi Anda dapat menggunakan fitur bahasa pemrograman Anda untuk menyaring respons.

Misalnya, AWS CLI perintah berikut hanya mendapatkan alias untukKunci yang dikelola AWS.

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

Perintah berikut hanya mendapatkan alias `access-key`. Nama alias peka terhadap huruf besar-kecil.

```
$ 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"
    }
]
```

Memperbarui alias

Karena alias adalah sumber daya independen, Anda dapat mengubah kunci KMS yang terkait dengan alias. Misalnya, jika `test-key` alias dikaitkan dengan satu kunci KMS, Anda dapat menggunakan UpdateAliasoperasi untuk mengaitkannya dengan kunci KMS yang berbeda. Ini adalah salah satu dari beberapa cara untuk memutar kunci KMS secara manual tanpa mengubah materi utamanya. Anda juga dapat memperbarui kunci KMS sehingga aplikasi yang menggunakan satu kunci KMS untuk sumber daya baru sekarang menggunakan kunci KMS yang berbeda.

Anda tidak dapat memperbarui alias di konsol AWS KMS. Juga, Anda tidak dapat menggunakan `UpdateAlias` (atau operasi lainnya) untuk mengubah nama alias. Untuk mengubah nama alias, hapus alias saat ini dan kemudian buat alias baru untuk kunci KMS.

Saat Anda memperbarui alias, kunci KMS saat ini dan kunci KMS baru harus tipe yang sama (simetris atau asimetris atau HMAC). Mereka juga harus memiliki penggunaan kunci yang sama (`ENCRYPT_DECRYPT`atau `SIGN_VERIFY` atau GENERATE_VERIFY_MAC). Pembatasan ini mencegah kesalahan kriptografi dalam kode yang menggunakan alias.

Contoh berikut dimulai dengan menggunakan ListAliasesoperasi untuk menunjukkan bahwa `test-key` alias saat ini dikaitkan dengan kunci KMS. `1234abcd-12ab-34cd-56ef-1234567890ab`

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/test-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        }
    ]
}
```

Selanjutnya, ia menggunakan `UpdateAlias` operasi untuk mengubah kunci KMS yang terkait dengan `test-key` alias ke kunci KMS. `0987dcba-09fe-87dc-65ba-ab0987654321` Anda tidak perlu menentukan kunci KMS yang saat ini terkait, hanya kunci KMS baru (“target”). Nama alias peka terhadap huruf besar-kecil.

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

Untuk memverifikasi bahwa alias sekarang dikaitkan dengan kunci KMS target, gunakan `ListAliases` operasi lagi. Perintah AWS CLI ini menggunakan parameter `--query` untuk hanya mendapatkan alias `test-key`. Bidang `TargetKeyId` dan `LastUpdatedDate` diperbarui.

```
$ 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
    }
]
```

Menghapus alias

Anda dapat menghapus alias di AWS KMS konsol atau dengan menggunakan DeleteAliasoperasi. Sebelum menghapus alias, pastikan bahwa itu tidak digunakan. Meskipun menghapus alias tidak memengaruhi kunci KMS terkait, itu mungkin menimbulkan masalah untuk aplikasi apa pun yang menggunakan alias. Jika Anda menghapus alias secara tidak sengaja, Anda dapat membuat alias baru dengan nama yang sama dan mengaitkannya dengan kunci KMS yang sama atau berbeda.

Jika Anda menghapus kunci KMS, semua alias yang terkait dengan kunci KMS tersebut akan dihapus.

Menghapus alias (konsol)

Untuk menghapus alias di AWS KMS konsol, gunakan tab Alias pada halaman detail untuk tombol KMS. Anda dapat menghapus beberapa alias untuk kunci KMS sekaligus.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. Anda tidak dapat mengelola alias untuk Kunci yang dikelola AWS atauKunci milik AWS.

4. Dalam tabel, pilih ID kunci atau alias kunci KMS. Kemudian, pada halaman detail kunci KMS, pilih tab Alias.

   Jika kunci KMS memiliki beberapa alias, kolom Alias dalam tabel menampilkan satu alias dan ringkasan alias, seperti (+ n lebih). Memilih ringkasan alias akan membawa Anda langsung ke tab Alias pada halaman detail kunci KMS.
5. Pada tab Alias, pilih kotak centang di samping alias yang ingin Anda hapus. Kemudian pilih Hapus.

Hapus alias (API AWS KMS)

Untuk menghapus alias, gunakan DeleteAliasoperasi. Operasi ini menghapus satu alias pada satu waktu. Nama alias peka terhadap huruf besar-kecil dan harus didahului oleh prefiks `alias/`.

Misalnya, perintah berikut menghapus alias `test-key`. Perintah ini tidak memberikan output apa pun.

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

Untuk memverifikasi bahwa alias dihapus, gunakan ListAliasesoperasi. Perintah berikut menggunakan parameter `--query` di AWS CLI untuk hanya mendapatkan alias `test-key`. Tanda kurung kosong dalam respons menunjukkan bahwa respons `ListAliases` tidak menyertakan alias `test-key`. Untuk menghilangkan tanda kurung, gunakan parameter dan nilai `--output text`.

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

Menggunakan alias dalam aplikasi Anda

Anda dapat menggunakan alias untuk mewakili kunci KMS dalam kode aplikasi Anda. `KeyId`Parameter dalam operasi AWS KMS kriptografi, DescribeKey, dan GetPublicKeymenerima nama alias atau alias ARN.

Misalnya, `GenerateDataKey` perintah berikut menggunakan nama alias (`alias/finance`) untuk mengidentifikasi kunci KMS. Nama alias adalah nilai dari parameter `KeyId`.

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

Jika kunci KMS berbedaAkun AWS, Anda harus menggunakan kunci ARN atau alias ARN dalam operasi ini. Saat menggunakan alias ARN, ingatlah bahwa alias untuk kunci KMS didefinisikan dalam akun yang memiliki kunci KMS dan mungkin berbeda di setiap Wilayah. Untuk bantuan menemukan alias ARN, lihat Menemukan nama alias dan ARN alias.

Misalnya, `GenerateDataKey` perintah berikut menggunakan kunci KMS yang tidak ada di akun pemanggil. `ExampleAlias`Alias dikaitkan dengan kunci KMS di akun dan Wilayah yang ditentukan.

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

Salah satu penggunaan alias yang paling kuat adalah pada aplikasi yang berjalan dalam beberapa Wilayah AWS. Misalnya, Anda mungkin memiliki aplikasi global yang menggunakan kunci KMS asimetris RSA untuk penandatanganan dan verifikasi.

- Di AS Barat (Oregon) (us-west-2), Anda dapat menggunakan `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- Di Eropa (Frankfurt) (eu-central-1), Anda dapat menggunakan `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321`
- Di Asia Pasifik (Singapura) (ap-southeast-1), Anda ingin menggunakan `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d`.

Anda dapat membuat versi aplikasi yang berbeda di setiap Wilayah atau menggunakan kamus atau pernyataan switch untuk memilih kunci KMS yang tepat untuk setiap Wilayah. Tetapi jauh lebih mudah untuk membuat alias dengan nama alias yang sama di setiap Wilayah. Ingat bahwa nama alias peka terhadap huruf besar-kecil.

```
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
```

Kemudian, gunakan alias dalam kode Anda. Ketika kode Anda berjalan di setiap Wilayah, alias akan merujuk ke kunci KMS terkait di Wilayah tersebut. Sebagai contoh, kode ini memanggil operasi Tanda tangan dengan nama alias.

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

Namun, ada risiko bahwa alias mungkin dihapus atau diperbarui untuk dikaitkan dengan kunci KMS yang berbeda. Dalam hal ini, upaya aplikasi untuk memverifikasi tanda tangan menggunakan nama alias akan gagal, dan Anda mungkin perlu membuat ulang atau memperbarui alias.

Untuk mengurangi risiko ini, berhati-hatilah dengan memberikan prinsipal izin untuk mengelola alias yang Anda gunakan dalam aplikasi Anda. Untuk rincian selengkapnya, lihat Mengontrol akses ke alias.

Ada beberapa solusi lain untuk aplikasi yang mengenkripsi data dalam beberapa Wilayah AWS, termasuk AWS Encryption SDK.

Mengontrol akses ke alias

Saat Anda membuat atau mengubah alias, Anda memengaruhi alias dan kunci KMS yang terkait. Oleh karena itu, kepala sekolah yang mengelola alias harus memiliki izin untuk memanggil operasi alias pada alias dan pada semua kunci KMS yang terpengaruh. Anda dapat memberikan izin ini dengan menggunakan kebijakan utama, Kebijakan IAM dan hibah.

> **Note**
>
> Berhati-hatilah saat memberikan izin prinsipal untuk mengelola tanda dan alias. Mengubah tag atau alias dapat mengizinkan atau menolak izin ke kunci yang dikelola pelanggan. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan alias untuk mengontrol akses ke tombol KMS.

Untuk informasi tentang mengontrol akses ke semua operasi AWS KMS, lihat Referensi izin.

Izin untuk membuat dan mengelola alias bekerja sebagai berikut.

km: CreateAlias

Untuk membuat alias, prinsipal memerlukan izin berikut untuk alias dan kunci KMS terkait.

- `kms:CreateAlias` untuk alias. Memberikan izin ini dalam kebijakan IAM yang melekat pada prinsipal yang diizinkan untuk membuat alias.

  Contoh pernyataan kebijakan berikut menentukan alias tertentu dalam elemen `Resource`. Tetapi Anda dapat mendaftar beberapa alias ARN atau menentukan pola alias, seperti "tes*". Anda juga dapat menentukan nilai `Resource` dari `"*"` untuk mengizinkan prinsipal untuk membuat alias apa pun di akun dan Wilayah. Izin untuk membuat alias juga dapat dimasukkan dalam izin `kms:Create*` untuk semua sumber daya di akun dan Wilayah.

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

- `kms:CreateAlias`untuk kunci KMS. Izin ini harus disediakan dalam kebijakan kunci atau kebijakan IAM yang didelegasikan dari kebijakan kunci.

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

Anda dapat menggunakan tombol kondisi untuk membatasi kunci KMS yang dapat Anda kaitkan dengan alias. Misalnya, Anda dapat menggunakan kms: KeySpec condition key untuk mengizinkan prinsipal membuat alias hanya pada kunci KMS asimetris. Untuk daftar lengkap kunci kondisi yang dapat Anda gunakan untuk membatasi `kms:CreateAlias` izin pada sumber daya kunci KMS, lihatAWS KMS izin.

km: ListAliases

Untuk mencantumkan alias di akun dan Wilayah, prinsipal harus memiliki izin `kms:ListAliases` dalam kebijakan IAM. Karena kebijakan ini tidak terkait dengan kunci KMS atau sumber alias tertentu, nilai elemen sumber daya dalam kebijakan harus. `"*"`

Misalnya, pernyataan kebijakan IAM berikut memberikan izin utama untuk mencantumkan semua kunci dan alias KMS di akun dan Wilayah.

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

km: UpdateAlias

Untuk mengubah kunci KMS yang terkait dengan alias, prinsipal membutuhkan tiga elemen izin: satu untuk alias, satu untuk kunci KMS saat ini, dan satu untuk kunci KMS baru.

Misalnya, Anda ingin mengubah `test-key` alias dari kunci KMS dengan ID kunci 1234abcd-12ab-34cd-56ef-1234567890ab ke kunci KMS dengan ID kunci 0987dcba-09fe-87dc-65ba-ab0987654321. Dalam hal ini, sertakan pernyataan kebijakan yang serupa dengan contoh pada bagian ini.

- `kms:UpdateAlias` untuk alias. Anda memberikan izin ini dalam kebijakan IAM yang melekat pada prinsipal. Kebijakan IAM berikut menentukan alias tertentu. Tetapi Anda dapat mendaftar beberapa alias ARN atau menentukan pola alias, seperti `"test*"`. Anda juga dapat menentukan

nilai `Resource` dari `"*"` untuk mengizinkan prinsipal untuk memperbarui alias apa pun di akun dan Wilayah.

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

- `kms:UpdateAlias`untuk kunci KMS yang saat ini dikaitkan dengan alias. Izin ini harus disediakan dalam kebijakan kunci atau kebijakan IAM yang didelegasikan dari kebijakan kunci.

```
{
  "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`untuk kunci KMS yang diasosiasikan operasi dengan alias. Izin ini harus disediakan dalam kebijakan kunci atau kebijakan IAM yang didelegasikan dari kebijakan kunci.

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

Anda dapat menggunakan tombol kondisi untuk membatasi salah satu atau kedua kunci KMS dalam suatu `UpdateAlias` operasi. Misalnya, Anda dapat menggunakan kms: ResourceAliases condition key untuk mengizinkan prinsipal memperbarui alias hanya ketika kunci KMS target sudah memiliki alias tertentu. Untuk daftar lengkap kunci kondisi yang dapat Anda gunakan untuk membatasi `kms:UpdateAlias` izin pada sumber daya kunci KMS, lihatAWS KMS izin.

km: DeleteAlias

Untuk menghapus alias, prinsipal memerlukan izin untuk alias dan untuk kunci KMS terkait.

Seperti biasa, Anda harus berhati-hati saat memberikan izin kepada pengguna utama untuk menghapus sumber daya. Namun, menghapus alias tidak berpengaruh pada kunci KMS terkait. Meskipun mungkin menyebabkan kegagalan dalam aplikasi yang bergantung pada alias, jika Anda tidak sengaja menghapus alias, Anda dapat membuatnya kembali.

- `kms:DeleteAlias` untuk alias. Berikan izin ini dalam kebijakan IAM yang melekat pada prinsipal yang diizinkan untuk menghapus alias.

  Contoh pernyataan kebijakan berikut menentukan alias dalam elemen. `Resource` Tapi Anda dapat membuat daftar beberapa ARN alias atau menentukan pola alias, seperti `"test*"`, Anda juga dapat menentukan nilai `Resource` dari `"*"` untuk mengizinkan prinsipal menghapus alias apa pun di akun dan Wilayah.

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

- `kms:DeleteAlias`untuk kunci KMS terkait. Izin ini harus disediakan dalam kebijakan kunci atau kebijakan IAM yang didelegasikan dari kebijakan kunci.

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

Membatasi izin alias

Anda dapat menggunakan tombol kondisi untuk membatasi izin alias ketika sumber daya adalah kunci KMS. Misalnya, kebijakan IAM berikut memungkinkan operasi alias pada kunci KMS di akun dan Wilayah tertentu. Namun, ia menggunakan kunci KeyOrigin kondisi kms: untuk lebih membatasi izin ke kunci KMS dengan materi kunci dari. AWS KMS

Untuk daftar lengkap kunci kondisi yang dapat Anda gunakan untuk membatasi izin alias pada sumber daya kunci KMS, lihat. AWS KMS izin

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

Anda tidak dapat menggunakan kunci kondisi dalam pernyataan kebijakan di mana sumber daya adalah alias. Untuk membatasi alias yang dapat dikelola oleh prinsipal, gunakan nilai dari elemen `Resource` dari pernyataan kebijakan IAM yang mengontrol akses ke alias. Sebagai contoh,

pernyataan kebijakan berikut mengizinkan prinsipal untuk membuat, memperbarui, atau menghapus alias apa pun di Akun AWS dan Wilayah kecuali alias dimulai dengan `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*"
}
```

Menggunakan alias untuk mengontrol akses ke tombol KMS

Anda dapat mengontrol akses ke kunci KMS berdasarkan alias yang terkait dengan kunci KMS. Untuk melakukannya, gunakan tombol ResourceAliases kondisi kms: RequestAlias dan kms:. Fitur ini adalah bagian dari dukungan AWS KMS untuk kontrol akses berbasis atribut (ABAC).

Kunci `kms:RequestAlias` kondisi memungkinkan atau menolak akses ke kunci KMS berdasarkan alias dalam permintaan. Kunci `kms:ResourceAliases` kondisi memungkinkan atau menolak akses ke kunci KMS berdasarkan alias yang terkait dengan kunci KMS.

Fitur-fitur ini tidak memungkinkan Anda untuk mengidentifikasi kunci KMS dengan menggunakan alias dalam `resource` elemen pernyataan kebijakan. Ketika alias adalah nilai `resource` elemen, kebijakan berlaku untuk sumber daya alias, bukan untuk kunci KMS apa pun yang mungkin terkait dengannya.

> **Note**
>
> Mungkin diperlukan waktu hingga lima menit untuk perubahan tag dan alias untuk memengaruhi otorisasi kunci KMS. Perubahan terbaru mungkin terlihat dalam operasi API sebelum mempengaruhi otorisasi.

Saat menggunakan alias untuk mengontrol akses ke kunci KMS, pertimbangkan hal berikut:

- Gunakan alias untuk memperkuat praktik terbaik dari akses dengan keistimewaan terkecil. Berikan kepala sekolah IAM hanya izin yang mereka butuhkan hanya untuk kunci KMS yang harus mereka gunakan atau kelola. Misalnya, gunakan alias untuk mengidentifikasi kunci KMS yang digunakan untuk proyek. Kemudian berikan izin tim proyek untuk hanya menggunakan kunci KMS dengan alias proyek.
- Berhati-hatilah tentang memberikan prinsipal izin `kms:CreateAlias`, `kms:UpdateAlias`, atau `kms:DeleteAlias` yang memungkinkan mereka menambahkan, mengedit, dan menghapus alias. Saat Anda menggunakan alias untuk mengontrol akses ke kunci KMS, mengubah alias dapat memberikan izin kepada prinsipal untuk menggunakan kunci KMS yang tidak diizinkan untuk digunakan. Ini juga dapat menolak akses ke kunci KMS yang diperlukan oleh kepala sekolah lain untuk melakukan pekerjaan mereka.
- Tinjau prinsipal di Akun AWS Anda yang saat ini memiliki izin untuk mengelola alias dan menyesuaikan izin, jika perlu. Administrator kunci yang tidak memiliki izin untuk mengubah kebijakan kunci atau membuat hibah dapat mengontrol akses ke kunci KMS jika mereka memiliki izin untuk mengelola alias.

  Sebagai contoh, konsol kebijakan kunci default untuk administrator kunci termasuk izin `kms:CreateAlias`, `kms:DeleteAlias`, dan `kms:UpdateAlias`. Kebijakan IAM mungkin memberikan izin alias untuk semua kunci KMS di Anda. Akun AWS Misalnya, kebijakan AWSKeyManagementServicePowerUserterkelola memungkinkan prinsipal untuk membuat, menghapus, dan mencantumkan alias untuk semua kunci KMS tetapi tidak memperbaruinya.
- Sebelum menetapkan kebijakan yang bergantung pada alias, tinjau alias pada kunci KMS di Anda. Akun AWS Pastikan bahwa kebijakan Anda hanya berlaku untuk alias yang ingin Anda sertakan. Gunakan CloudTrail log dan CloudWatch alarm untuk mengingatkan Anda tentang perubahan alias yang mungkin memengaruhi akses ke kunci KMS Anda. Selain itu, ListAliasestanggapannya mencakup tanggal pembuatan dan tanggal pembaruan terakhir untuk setiap alias.

- Kondisi kebijakan alias menggunakan pencocokan pola; mereka tidak terikat pada instans tertentu dari alias. Kebijakan yang menggunakan kunci kondisi berbasis alias memengaruhi semua alias baru dan yang sudah ada yang cocok dengan pola. Jika Anda menghapus dan membuat ulang alias yang cocok dengan kondisi kebijakan, kondisi berlaku untuk alias baru, seperti halnya pada tanda lama.

Kunci kondisi `kms:RequestAlias` bergantung pada alias yang ditentukan secara eksplisit dalam permintaan operasi. Kunci `kms:ResourceAliases` kondisi tergantung pada alias yang terkait dengan kunci KMS, meskipun tidak muncul dalam permintaan.

km: RequestAlias

Izinkan atau tolak akses ke kunci KMS berdasarkan alias yang mengidentifikasi kunci KMS dalam permintaan. Anda dapat menggunakan kms: RequestAlias condition key dalam kebijakan kunci atau kebijakan IAM. Ini berlaku untuk operasi yang menggunakan alias untuk mengidentifikasi kunci KMS dalam permintaan, yaitu operasi kriptografi,, DescribeKeydan. GetPublicKey Ini tidak berlaku untuk operasi alias, seperti CreateAliasatau DeleteAlias.

Dalam kunci kondisi, tentukan nama alias atau pola nama alias. Anda tidak dapat menentukan ARN alias.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan prinsipal untuk menggunakan operasi yang ditentukan pada kunci KMS. Izin hanya efektif ketika permintaan menggunakan alias yang mencakup `alpha` untuk mengidentifikasi kunci KMS.

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/alpha-developer"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:RequestAlias": "alias/*alpha*"
    }
```

```
  }
}
```

Contoh permintaan berikut dari prinsipal terotorisasi akan memenuhi kondisi tersebut. Namun, permintaan yang menggunakan ID kunci, ARN kunci, atau alias yang berbeda tidak akan memenuhi kondisi, bahkan jika nilai-nilai ini mengidentifikasi kunci KMS yang sama.

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

km: ResourceAliases

Izinkan atau tolak akses ke kunci KMS berdasarkan alias yang terkait dengan kunci KMS, bahkan jika alias tidak digunakan dalam permintaan. Kunci ResourceAliases kondisi kms: memungkinkan Anda menentukan pola alias atau alias, seperti`alias/test*`, sehingga Anda dapat menggunakannya dalam kebijakan IAM untuk mengontrol akses ke beberapa kunci KMS di Wilayah yang sama. Ini berlaku untuk setiap AWS KMS operasi yang menggunakan kunci KMS.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal mengelola rotasi kunci otomatis pada kunci KMS menjadi dua. Akun AWS Namun, izin hanya berlaku untuk kunci KMS yang terkait dengan alias yang dimulai dengan. `restricted`

```
{
  "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*"
        }
```

```
      }
    }
  ]
}
```

Kondisi `kms:ResourceAliases` adalah kondisi sumber daya, bukan permintaan. Dengan demikian, permintaan yang tidak menentukan alias masih dapat memenuhi kondisi.

Contoh permintaan berikut, yang menentukan alias yang cocok, memenuhi kondisi.

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

Namun, contoh permintaan berikut juga memenuhi kondisi, asalkan kunci KMS yang ditentukan memiliki alias yang dimulai dengan`restricted`, bahkan jika alias itu tidak digunakan dalam permintaan.

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

Menemukan alias dalam log AWS CloudTrail

Anda dapat menggunakan alias untuk mewakili AWS KMS key dalam operasi AWS KMS API. Ketika Anda melakukannya, alias dan kunci ARN dari kunci KMS dicatat dalam entri log AWS CloudTrail untuk acara tersebut. Alias muncul dalam bidang `requestParameters`. ARN kunci muncul dalam bidang `resources`. Hal ini berlaku bahkan ketika AWS layanan menggunakan Kunci yang dikelola AWS di akun Anda.

Misalnya, GenerateDataKeypermintaan berikut menggunakan `project-key` alias untuk mewakili kunci KMS.

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

Ketika permintaan ini dicatat dalam CloudTrail log, entri log mencakup alias dan kunci ARN dari kunci KMS aktual yang digunakan.

```
{
    "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"
}
```


Untuk detail tentang AWS KMS operasi pencatatan di CloudTrail log, lihatLogging panggilan AWS KMS API dengan AWS CloudTrail.

Melihat kunci

Anda dapat menggunakan AWS Management Consoleatau AWS Key Management Service(AWS KMS) API untuk melihat AWS KMS keys di setiap akun dan Wilayah, termasuk kunci KMS yang Anda kelola dan kunci KMS yang dikelola oleh. AWS

Topik

- Melihat tombol KMS di konsol

- Melihat kunci KMS dengan API
- Melihat konfigurasi kriptografi kunci KMS
- Menemukan ID kunci dan kunci ARN
- Menemukan nama alias dan ARN alias

Melihat tombol KMS di konsol

Di dalamAWS Management Console, Anda dapat melihat daftar kunci KMS Anda di akun dan Wilayah dan detail tentang setiap kunci KMS.

> **Note**
>
> AWS KMSKonsol menampilkan kunci KMS yang Anda memiliki izin untuk melihat di akun dan Wilayah Anda. Kunci KMS di lain Akun AWS tidak muncul di konsol, bahkan jika Anda memiliki izin untuk melihat, mengelola, dan menggunakannya. Untuk melihat kunci KMS di akun lain, gunakan DescribeKeyoperasi.

Topik

- Menavigasi ke tabel kunci
- Menavigasi ke detail kunci
- Menyortir dan memfilter kunci KMS Anda
- Menampilkan detail kunci KMS
- Menyesuaikan tabel kunci KMS Anda

Menavigasi ke tabel kunci

AWS KMS keysDi setiap akun dan Wilayah ditampilkan dalam tabel. Ada tabel terpisah untuk kunci KMS yang Anda buat dan kunci KMS yang dibuat AWS layanan untuk Anda.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di

panel navigasi pilih kunci yang dikelola AWS. Untuk informasi tentang berbagai jenis kunci KMS, lihat AWS KMS keys.

> **Tip**
>
> Untuk melihat alias Kunci yang dikelola AWSyang hilang, gunakan halaman kunci yang dikelola Pelanggan.
> Konsol AWS KMS juga menampilkan penyimpanan kunci kustom di akun dan Wilayah. Kunci KMS yang Anda buat di toko kunci kustom muncul di halaman kunci yang dikelola Pelanggan. Untuk informasi tentang penyimpanan kunci kustom, lihat Penyimpanan kunci kustom.

Menavigasi ke detail kunci

Ada halaman detail untuk setiap AWS KMS key akun dan Wilayah. Halaman detail menampilkan bagian konfigurasi Umum untuk kunci KMS dan menyertakan tab yang memungkinkan pengguna yang berwenang melihat dan mengelola konfigurasi Kriptografi dan kebijakan Kunci untuk kunci tersebut. Bergantung pada jenis kunci, halaman detail mungkin juga mencakup tab Alias, Materi kunci, Rotasi kunci, Kunci publik, Regionalitas, dan Tag.

Untuk menavigasi ke halaman detail utama untuk kunci KMS.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih kunci yang dikelola AWS. Untuk informasi tentang berbagai jenis kunci KMS, lihat AWS KMS key.
4. Untuk membuka halaman detail kunci, di tabel kunci, pilih ID kunci atau alias kunci KMS.

   Jika kunci KMS memiliki beberapa alias, ringkasan alias (+ n lebih) muncul di samping nama salah satu alias. Memilih ringkasan alias membawa Anda langsung ke tab Alias pada halaman detail kunci.

Menyortir dan memfilter kunci KMS Anda

Untuk membuatnya lebih mudah untuk menemukan kunci KMS Anda di konsol, Anda dapat mengurutkan dan memfilter tabel kunci.

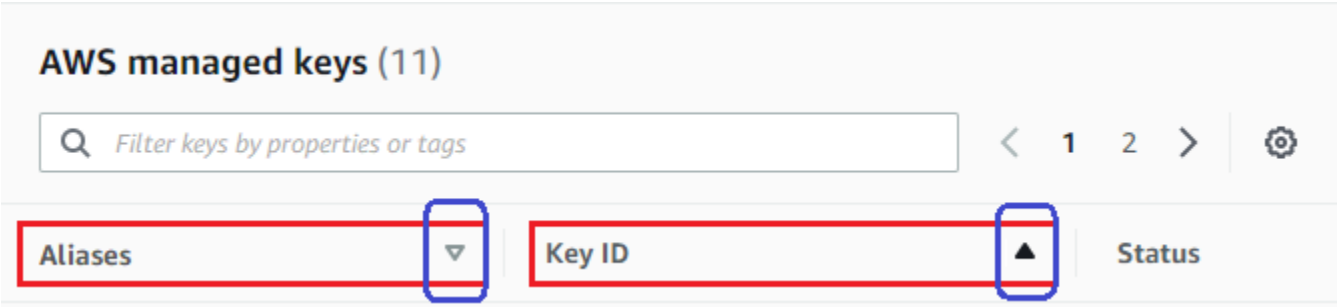
Urutkan

Anda dapat mengurutkan kunci KMS dalam urutan naik atau turun berdasarkan nilai kolomnya. Fitur ini mengurutkan semua kunci KMS dalam tabel, bahkan jika mereka tidak muncul di halaman tabel saat ini.

Kolom yang dapat diurutkan ditunjukkan dengan panah di samping nama kolom. Pada Kunci yang dikelola AWShalaman, Anda dapat mengurutkan berdasarkan Alias atau ID Kunci. Pada halaman Kunci terkelola pelanggan, Anda dapat mengurutkan berdasarkan Alias, ID Kunci, atau Tipe kunci.

Untuk mengurutkan dalam urutan menaik, pilih judul kolom sampai panah menunjuk ke atas. Untuk mengurutkan dalam urutan menurun, pilih judul kolom sampai panah menunjuk ke bawah. Anda dapat mengurutkan dengan hanya satu kolom pada satu waktu.

Misalnya, Anda dapat mengurutkan kunci KMS dalam urutan menaik berdasarkan ID kunci, bukan alias, yang merupakan default.



Saat Anda mengurutkan kunci KMS pada halaman kunci yang dikelola Pelanggan dalam urutan menaik menurut jenis Kunci, semua kunci asimetris ditampilkan sebelum semua kunci simetris.

Filter

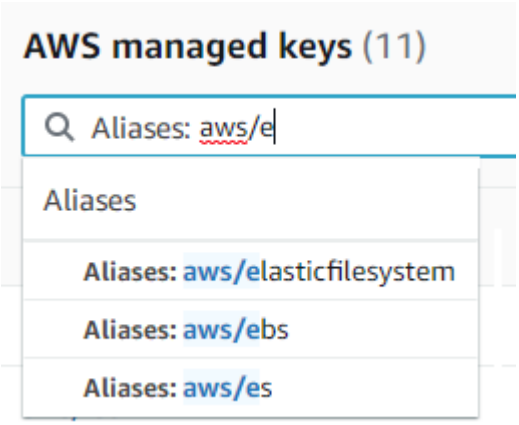
Anda dapat memfilter kunci KMS berdasarkan nilai properti atau tag mereka. Filter berlaku untuk semua kunci KMS dalam tabel, bahkan jika mereka tidak muncul di halaman tabel saat ini. Filter tidak peka huruf besar/kecil.

Properti yang dapat disaring tercantum dalam kotak filter. Pada Kunci yang dikelola AWShalaman, Anda dapat memfilter berdasarkan alias dan ID kunci. Pada halaman kunci terkelola Pelanggan, Anda dapat memfilter berdasarkan alias, ID kunci, dan properti tipe kunci, dan menurut tag.

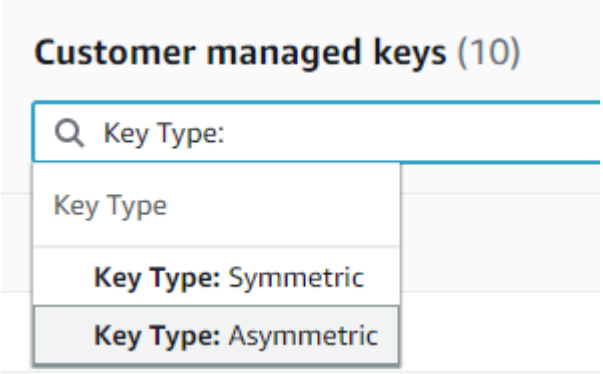
- Pada Kunci yang dikelola AWShalaman, Anda dapat memfilter berdasarkan alias dan ID kunci.
- Pada halaman Kunci terkelola pelanggan, Anda dapat memfilter berdasarkan tanda, atau berdasarkan alias, ID kunci, jenis kunci, atau properti regionalitas.

Untuk memfilter berdasarkan nilai properti, pilih filter, pilih nama properti, lalu pilih dari daftar nilai properti sebenarnya. Untuk memfilter berdasarkan tanda, pilih kunci tanda, lalu pilih dari daftar nilai tanda yang sebenarnya. Setelah memilih properti atau tag kunci, Anda juga dapat mengetik semua atau sebagian dari nilai properti atau nilai tanda. Anda akan melihat pratinjau hasil sebelum memilih.

Misalnya, untuk menampilkan tombol KMS dengan nama alias yang berisi`aws/e`, pilih kotak filter, pilih Alias, ketik`aws/e`, lalu tekan `Enter` atau `Return` tambahkan filter.

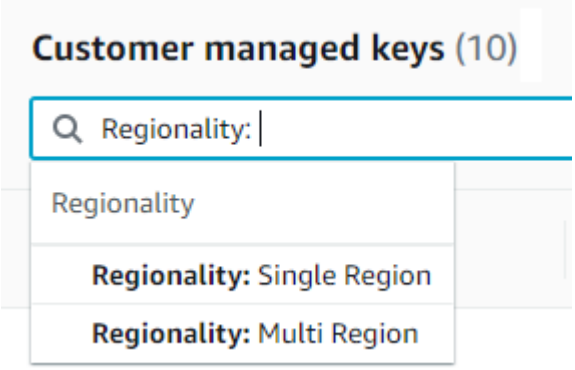


Untuk menampilkan hanya kunci KMS asimetris pada halaman Kunci yang dikelola Pelanggan, klik kotak filter, pilih Key type dan kemudian pilih Key type: Asymmetric. Opsi Asimetris hanya muncul ketika Anda memiliki kunci KMS asimetris dalam tabel. Untuk informasi selengkapnya tentang mengidentifikasi kunci KMS asimetris, lihat. Mengidentifikasi kunci KMS asimetris



Untuk hanya menampilkan kunci Multi-region, pada halaman Customer managed keys, pilih kotak filter, pilih Regionality dan kemudian pilih Regionality: Multi-Region. Opsi Multi-Region hanya

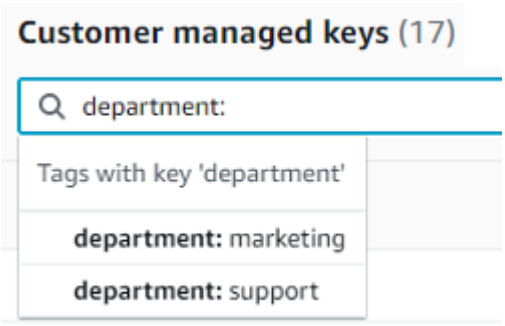
muncul ketika Anda memiliki kunci Multi-wilayah dalam tabel. Untuk informasi selengkapnya tentang mengidentifikasi kunci Multi-wilayah, lihatMelihat kunci multi-Wilayah.



Pemfilteran tanda sedikit berbeda. Untuk hanya menampilkan tombol KMS dengan tag tertentu, pilih kotak filter, pilih kunci tag, lalu pilih dari antara nilai tag yang sebenarnya. Anda juga dapat mengetik semua atau sebagian nilai tanda.

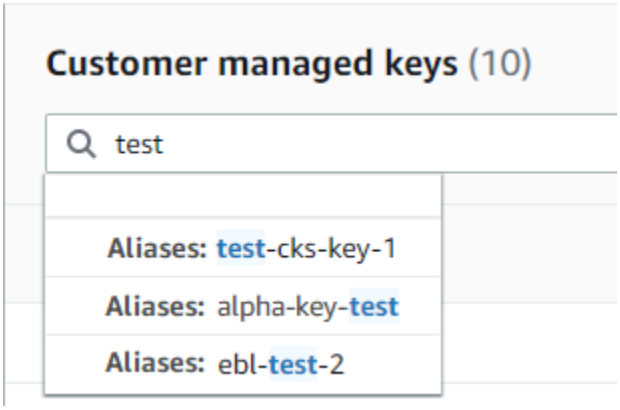
Tabel yang dihasilkan menampilkan semua kunci KMS dengan tag yang dipilih. Namun, itu tidak menampilkan tanda. Untuk melihat tag, pilih ID kunci atau alias tombol KMS dan pada halaman detailnya, pilih tab Tag. Tab muncul di bawah bagian Konfigurasi umum.

Filter ini membutuhkan kunci tag dan nilai tag. Itu tidak akan menemukan kunci KMS dengan mengetik hanya kunci tag atau hanya nilainya. Untuk memfilter tag berdasarkan semua atau sebagian kunci tag atau nilai, gunakan ListResourceTagsoperasi untuk mendapatkan tombol KMS yang diberi tag, lalu gunakan fitur pemfilteran bahasa pemrograman Anda. Sebagai contoh, lihat ListResourceTags: Dapatkan tag pada tombol KMS.

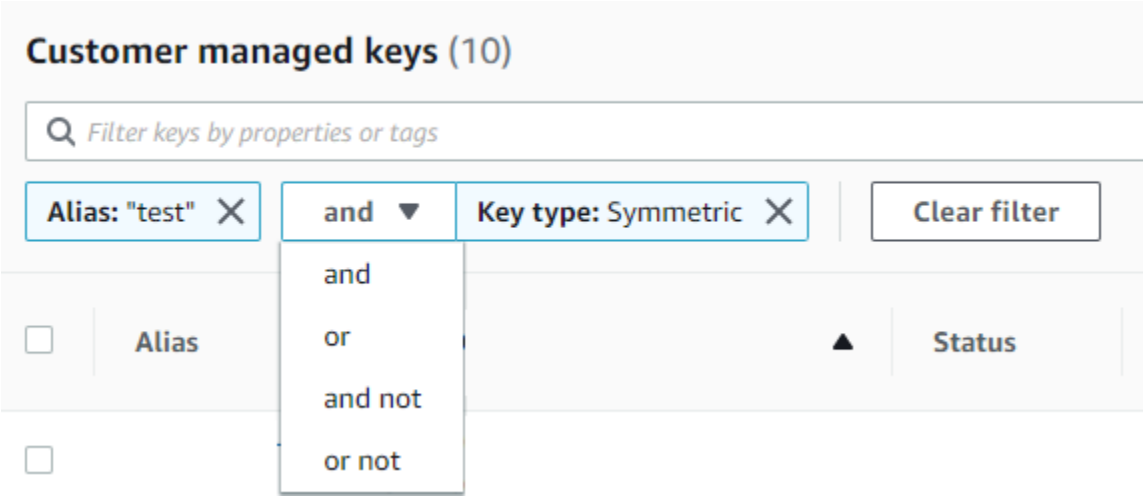


Untuk mencari teks, di kotak filter, ketik semua atau sebagian alias, ID kunci, jenis kunci, atau tombol tanda. (Setelah Anda memilih kunci tanda, Anda dapat mencari nilai tanda). Anda akan melihat pratinjau hasil sebelum memilih.

Misalnya, untuk menampilkan kunci KMS dengan `test` kunci tag atau properti yang dapat disaring, ketik kotak `test` filter. Pratinjau menunjukkan tombol KMS yang akan dipilih filter. Dalam kasus ini, `test` hanya muncul di properti Alias.



Anda dapat menggunakan beberapa filter pada saat bersamaan. Bila Anda menambahkan filter tambahan, Anda juga dapat memilih operator logis.



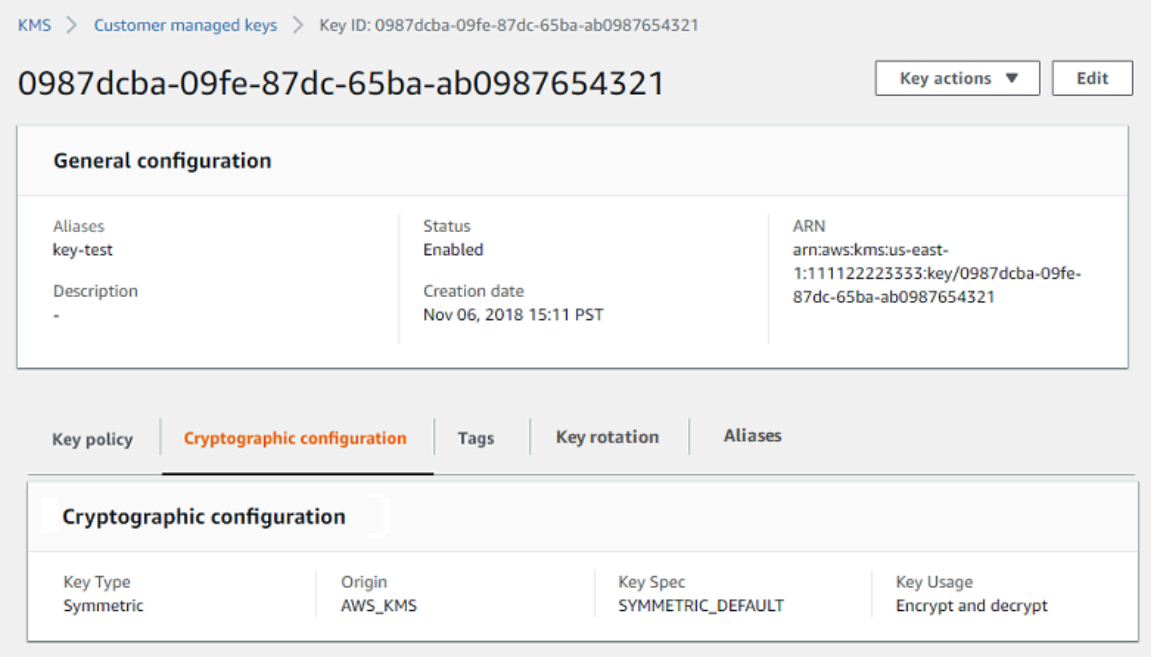
Menampilkan detail kunci KMS

Halaman detail untuk setiap tombol KMS menampilkan properti kunci KMS. Ini sedikit berbeda untuk berbagai jenis kunci KMS.

Untuk menampilkan informasi rinci tentang kunci KMS, pada Kunci yang dikelola AWSatau halaman kunci yang dikelola Pelanggan, pilih alias atau ID kunci kunci kunci KMS.

Halaman detail untuk kunci KMS mencakup bagian Konfigurasi Umum yang menampilkan properti dasar kunci KMS. Ini juga mencakup tab di mana Anda dapat melihat dan mengedit properti kunci KMS, seperti Kebijakan kunci, konfigurasi Kriptografi, Tag, Materi kunci (untuk kunci KMS dengan

bahan kunci yang diimpor), Rotasi kunci (untuk kunci KMS enkripsi simetris), Regionalitas (untuk kunci Multi-wilayah), dan kunci Publik (untuk kunci KMS asimetris).



Daftar berikut menjelaskan bidang dalam tampilan mendetail, termasuk bidang di tab. Beberapa bidang ini juga tersedia sebagai kolom dalam tampilan tabel.

Alias

Di mana: Tab alias

Nama yang ramah untuk kunci KMS. Anda dapat menggunakan alias untuk mengidentifikasi kunci KMS di konsol dan di beberapa AWS KMS API. Untuk detailnya, lihat Menggunakan alias.

Tab Alias menampilkan semua alias yang terkait dengan kunci KMS di dan Wilayah. Akun AWS

ARN

Di mana: Bagian konfigurasi umum

Nama Sumber Daya Amazon (ARN) dari kunci KMS. Nilai ini secara unik mengidentifikasi kunci KMS. Anda dapat menggunakannya untuk mengidentifikasi kunci KMS dalam operasi AWS KMS API.

Status koneksi

Menunjukkan apakah toko kunci khusus terhubung ke toko kunci pendukungnya. Bidang ini hanya muncul ketika kunci KMS dibuat di toko kunci khusus.

Untuk informasi tentang nilai di bidang ini, lihat ConnectionStatedi Referensi AWS KMS API.

Tanggal pembuatan

Di mana: Bagian konfigurasi umum

Tanggal dan waktu kunci KMS dibuat. Nilai ini ditampilkan dalam waktu lokal untuk perangkat. Zona waktu tidak tergantung pada Wilayah.

Tidak seperti Expiration, kreasi hanya mengacu pada kunci KMS, bukan materi utamanya.

ID klaster CloudHSM

Di mana: Tab konfigurasi kriptografi

ID cluster AWS CloudHSM cluster yang berisi materi kunci untuk kunci KMS. Bidang ini hanya muncul ketika kunci KMS dibuat di toko kunci khusus.

Jika Anda memilih ID klaster CloudHSM, itu akan membuka halaman Klaster dalam konsol AWS CloudHSM.

ID penyimpanan kunci kustom

Di mana: Tab konfigurasi kriptografi

ID toko kunci kustom yang berisi kunci KMS. Bidang ini hanya muncul ketika kunci KMS dibuat di toko kunci khusus.

Jika Anda memilih ID penyimpanan kunci kustom, ini membuka halaman Penyimpanan kunci khusus dalam konsol AWS KMS.

Nama penyimpanan kunci kustom

Di mana: Tab konfigurasi kriptografi

Nama toko kunci khusus yang berisi kunci KMS. Bidang ini hanya muncul ketika kunci KMS dibuat di toko kunci khusus.

Jenis toko kunci khusus

Di mana: Tab konfigurasi kriptografi

Menunjukkan apakah toko kunci kustom adalah toko AWS CloudHSM kunci atau toko kunci eksternal. Bidang ini hanya muncul ketika kunci KMS dibuat di toko kunci khusus.

Deskripsi

Di mana: Bagian konfigurasi umum

Deskripsi singkat dan opsional dari kunci KMS yang dapat Anda tulis dan edit. Untuk menambahkan atau memperbarui deskripsi kunci yang dikelola pelanggan, di atas Konfigurasi Umum, pilih Edit.

Algoritme enkripsi

Di mana: Tab konfigurasi kriptografi

Daftar algoritma enkripsi yang dapat digunakan dengan kunci KMS di. AWS KMS Bidang ini hanya muncul ketika Tipe kunci adalah Asimetris dan Penggunaan kunci adalah Enkripsi dan Dekripsi. Untuk informasi tentang algoritme enkripsi yang didukung AWS KMS, lihat Spesifikasi kunci SYMMETRIC_DEFAULT dan Spesifikasi kunci RSA untuk enkripsi dan dekripsi.

Tanggal kedaluwarsa

Di mana: Tab material kunci

Tanggal dan waktu ketika materi kunci untuk kunci KMS kedaluwarsa. Bidang ini hanya muncul untuk kunci KMS dengan bahan kunci yang diimpor, yaitu ketika Origin adalah Eksternal dan kunci KMS memiliki materi kunci yang kedaluwarsa.

ID kunci eksternal

Di mana: Tab konfigurasi kriptografi

ID kunci eksternal yang dikaitkan dengan kunci KMS di penyimpanan kunci eksternal. Bidang ini hanya muncul untuk kunci KMS di penyimpanan kunci eksternal.

Status kunci eksternal

Di mana: Tab konfigurasi kriptografi

Status terbaru yang dilaporkan proxy penyimpanan kunci eksternal untuk kunci eksternal yang terkait dengan kunci KMS. Bidang ini hanya muncul untuk kunci KMS di penyimpanan kunci eksternal.

Penggunaan kunci eksternal

Di mana: Tab konfigurasi kriptografi

Operasi kriptografi yang diaktifkan pada kunci eksternal yang terkait dengan kunci KMS. Bidang ini hanya muncul untuk kunci KMS di penyimpanan kunci eksternal.

Kebijakan kunci

Di mana: Tab kebijakan kunci

Mengontrol akses ke kunci KMS bersama dengan kebijakan dan hibah IAM. Setiap kunci KMS memiliki satu kebijakan utama. Ini adalah satu-satunya elemen otorisasi wajib. Untuk mengubah kebijakan kunci kunci yang dikelola pelanggan, pada tab Kebijakan kunci, pilih Edit. Untuk rincian selengkapnya, lihat the section called “Kebijakan utama”.

Rotasi kunci

Di mana: Tab rotasi kunci

Mengaktifkan dan menonaktifkan rotasi otomatis bahan utama dalam kunci KMS yang dikelola pelanggan. Untuk mengubah status rotasi kunci dari kunci yang dikelola pelanggan, gunakan kotak centang pada tab Rotasi kunci.

Anda tidak dapat mengaktifkan atau menonaktifkan rotasi materi kunci dalam file Kunci yang dikelola AWS. Kunci yang dikelola AWSsecara otomatis diputar setiap tahun.

Spesifikasi kunci

Di mana: Tab konfigurasi kriptografi

Jenis bahan kunci dalam kunci KMS. AWS KMSmendukung kunci KMS enkripsi simetris (SYMMETRIC_DEFAULT), kunci HMAC KMS dengan panjang yang berbeda, kunci KMS untuk kunci RSA dengan panjang yang berbeda, dan kunci kurva elips dengan kurva yang berbeda. Untuk rincian selengkapnya, lihat Spesifikasi kunci.

Tipe kunci

Di mana: Tab konfigurasi kriptografi

Menunjukkan apakah kunci KMS adalah Simetris atau Asimetris.

Penggunaan kunci

Di mana: Tab konfigurasi kriptografi

Menunjukkan apakah kunci KMS dapat digunakan untuk Enkripsi dan dekripsi, Menandatangani dan memverifikasi atau Menghasilkan dan memverifikasi MAC. Untuk rincian selengkapnya, lihat Penggunaan kunci.

asal

Di mana: Tab konfigurasi kriptografi

Sumber bahan utama untuk kunci KMS. Nilai yang valid adalah:

- AWS KMSuntuk bahan utama yang AWS KMS menghasilkan

- AWS CloudHSMuntuk kunci KMS di toko AWS CloudHSMkunci
- Eksternal untuk bahan kunci impor (BYOK)
- Penyimpanan kunci eksternal untuk kunci KMS di toko kunci eksternal

Algoritma MAC

Di mana: Tab konfigurasi kriptografi

Daftar algoritma MAC yang dapat digunakan dengan kunci HMAC KMS di. AWS KMS Bidang ini hanya muncul ketika spesifikasi Kunci adalah spesifikasi kunci HMAC (HMAC_*). Untuk informasi tentang algoritma MAC yang AWS KMS mendukung, lihatSpesifikasi utama untuk kunci HMAC KMS.

Kunci utama

Di mana: Tab regionalitas

Menunjukkan bahwa kunci KMS ini adalah kunci utama Multi-wilayah. Pengguna yang diotorisasi dapat menggunakan bagian ini untuk mengubah kunci primer ke kunci multi-wilayah terkait yang berbeda. Bidang ini hanya muncul ketika kunci KMS adalah kunci primer Multi-wilayah.

Kunci publik

Di mana: Tab kunci publik

Menampilkan kunci publik dari kunci KMS asimetris. Pengguna yang diotorisasi dapat menggunakan tab ini untuk menyalin dan mengunduh kunci publik.

Regionalitas

Di mana: Bagian konfigurasi umum dan Tab regionalitas

Menunjukkan apakah kunci KMS adalah kunci wilayah Tunggal, kunci primer Multi-wilayah, atau kunci replika Multi-wilayah. Bidang ini hanya muncul ketika kunci KMS adalah kunci Multi-wilayah.

Kunci Multi-wilayah yang terkait

Di mana: Tab regionalitas

Menampilkan semua kunci primer dan replika Multi-wilayah terkait, kecuali untuk kunci KMS saat ini. Bidang ini hanya muncul ketika kunci KMS adalah kunci Multi-wilayah.

Di bagian Kunci Multi-wilayah yang terkait dari kunci primer, pengguna yang diotorisasi dapat membuat kunci replika baru.

Kunci replika

Di mana: Tab regionalitas

Menunjukkan bahwa kunci KMS ini adalah kunci replika Multi-wilayah. Bidang ini hanya muncul ketika kunci KMS adalah kunci replika Multi-wilayah.

Algoritme penandatanganan

Di mana: Tab konfigurasi kriptografi

Daftar algoritma penandatanganan yang dapat digunakan dengan kunci KMS di. AWS KMS Bidang ini hanya muncul ketika Tipe kunci adalah Asimetris dan Penggunaan kunci adalah Tanda tangan dan verifikasi. Untuk informasi tentang algoritme penandatanganan yang didukung AWS KMS, lihat Spesifikasi kunci RSA untuk penandatanganan dan verifikasi dan Spesifikasi kunci kurva elips.

Status

Di mana: Bagian konfigurasi umum

Status kunci dari kunci KMS. Anda dapat menggunakan kunci KMS dalam operasi kriptografi hanya ketika status Diaktifkan. Untuk penjelasan rinci tentang setiap status kunci KMS dan pengaruhnya terhadap operasi yang dapat Anda jalankan pada tombol KMS, lihat. Status AWS KMS kunci kunci

Tag

Di mana: Tab Tanda

Pasangan nilai kunci opsional yang menggambarkan kunci KMS. Untuk menambah atau mengubah tag untuk kunci KMS, pada tab Tag, pilih Edit.

Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS

Menyesuaikan tabel kunci KMS Anda

Anda dapat menyesuaikan tabel yang muncul di halaman kunci yang dikelola Pelanggan Kunci yang dikelola AWSdan AWS Management Console sesuai dengan kebutuhan Anda. Anda dapat memilih kolom tabel, jumlah AWS KMS keys pada setiap halaman (Ukuran halaman), dan bungkus teks.

Konfigurasi yang Anda pilih akan disimpan saat mengonfirmasi dan diterapkan kembali setiap kali Anda membuka halaman.

Untuk menyesuaikan tabel kunci KMS Anda

1. Pada halaman Kunci yang dikelola AWSatau Kunci terkelola Pelanggan, pilih ikon pengaturan () di sudut kanan atas halaman.
2. Pada halaman Preferensi, pilih pengaturan yang Anda inginkan, kemudian pilih Konfirmasi.

Pertimbangkan untuk menggunakan pengaturan ukuran Halaman untuk menambah jumlah tombol KMS yang ditampilkan di setiap halaman, terutama jika Anda biasanya menggunakan perangkat yang mudah digulir.

Kolom data yang Anda tampilkan mungkin bervariasi tergantung pada tabel, peran pekerjaan Anda, dan jenis kunci KMS di akun dan Wilayah. Tabel berikut menawarkan beberapa konfigurasi yang disarankan. Untuk deskripsi kolom, lihat Menampilkan detail kunci KMS.

Konfigurasi tabel kunci KMS yang disarankan

Anda dapat menyesuaikan kolom yang muncul di tabel kunci KMS Anda untuk menampilkan informasi yang Anda butuhkan tentang kunci KMS Anda.

Kunci yang dikelola AWS

Secara default, Kunci yang dikelola AWStabel menampilkan kolom Alias, ID Kunci, dan Status. Kolom ini ideal untuk sebagian besar kasus penggunaan.

Kunci KMS enkripsi simetris

Jika Anda hanya menggunakan kunci KMS enkripsi simetris dengan materi kunci yang dihasilkan olehAWS KMS, kolom Alias, ID Kunci, Status, dan tanggal Pembuatan kemungkinan akan menjadi yang paling berguna.

Kunci Asymmetric KMS

Jika Anda menggunakan kunci KMS asimetris, selain kolom Alias, ID Kunci, dan Status, pertimbangkan untuk menambahkan kolom Key type, Key spec, dan Key usage. Kolom ini akan menunjukkan kepada Anda apakah kunci KMS simetris atau asimetris, jenis bahan kunci, dan apakah kunci KMS dapat digunakan untuk enkripsi atau penandatanganan.

Kunci HMAC KMS

Jika Anda menggunakan kunci HMAC KMS, selain kolom Alias, ID Kunci, dan Status, pertimbangkan untuk menambahkan kolom Spesifikasi Kunci dan penggunaan Kunci. Kolom ini akan menunjukkan kepada Anda apakah kunci KMS adalah kunci HMAC. Karena Anda tidak dapat mengurutkan kunci KMS berdasarkan spesifikasi kunci atau penggunaan kunci, gunakan alias dan tag untuk mengidentifikasi kunci HMAC Anda dan kemudian gunakan fitur filter AWS KMS konsol untuk memfilter berdasarkan alias atau tag.

Materi kunci yang diimpor

Jika Anda memiliki kunci KMS dengan bahan kunci yang diimpor, pertimbangkan untuk menambahkan kolom Tanggal Asal dan Kedaluwarsa. Kolom ini akan menunjukkan kepada Anda apakah materi kunci dalam kunci KMS diimpor atau dihasilkan oleh AWS KMS dan ketika materi kunci kedaluwarsa, jika sama sekali. Bidang Tanggal pembuatan menampilkan tanggal bahwa kunci KMS dibuat (tanpa materi kunci). Ini tidak mencerminkan karakteristik apa pun dari material kunci.

Kunci di penyimpanan kunci kustom

Jika Anda memiliki kunci KMS di toko kunci khusus, pertimbangkan untuk menambahkan kolom ID toko kunci Asal dan Kustom. Kolom ini menunjukkan bahwa kunci KMS ada di toko kunci khusus, menampilkan jenis toko kunci kustom, dan mengidentifikasi toko kunci kustom.

Kunci Multi-Wilayah

Jika Anda memiliki kunci Multi-wilayah, pertimbangkan untuk menambahkan kolom Regionalitas. Ini menunjukkan apakah kunci KMS adalah kunci wilayah Tunggal, kunci primer Multi-wilayah, atau kunci replika Multi-wilayah.

Melihat kunci KMS dengan API

Anda dapat menggunakan AWS Key Management Service(AWS KMS) API untuk melihat kunci KMS Anda. Bagian ini menunjukkan beberapa operasi yang mengembalikan rincian tentang kunci KMS yang ada. Contoh-contoh menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Topik

- ListKeys: Dapatkan ID dan ARN dari semua kunci KMS
- DescribeKey: Dapatkan informasi rinci tentang kunci KMS

- GetKeyPolicy: Dapatkan kebijakan kunci yang dilampirkan ke kunci KMS
- ListAliases: Dapatkan nama alias dan ARN untuk kunci KMS
- ListResourceTags: Dapatkan tag pada tombol KMS

ListKeys: Dapatkan ID dan ARN dari semua kunci KMS

ListKeysOperasi mengembalikan ID dan Nama Sumber Daya Amazon (ARN) dari semua kunci KMS di akun dan Wilayah.

Misalnya, panggilan ke `ListKeys` operasi ini mengembalikan ID dan ARN dari setiap kunci KMS di akun fiktif ini. Untuk contoh dalam beberapa bahasa pemrograman, lihat Mendapatkan ID kunci dan ARN kunci dari kunci KMS.

```
$ 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: Dapatkan informasi rinci tentang kunci KMS

DescribeKeyOperasi mengembalikan rincian tentang kunci KMS yang ditentukan. Untuk mengidentifikasi kunci KMS, gunakan ID kunci, kunciARN, nama alias, atau alias ARN.

Berbeda dengan ListKeysoperasi, yang hanya menampilkan kunci KMS di akun dan Wilayah pemanggil, pengguna yang berwenang dapat menggunakan `DescribeKey` operasi untuk mendapatkan detail tentang kunci KMS di akun lain.

> **Note**
>
> `DescribeKey`Tanggapan tersebut mencakup keduanya `KeySpec` dan `CustomerMasterKeySpec` anggota dengan nilai yang sama. `CustomerMasterKeySpec`Anggota tidak digunakan lagi.

Misalnya, panggilan ini untuk `DescribeKey` mengembalikan informasi tentang kunci KMS enkripsi simetris. Bidang dalam respons bervariasi dengan AWS KMS keyspesifikasi, status kunci, dan asal material utama. Untuk contoh dalam beberapa bahasa pemrograman, lihat Melihat sebuah 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"
        ]
    }
}
```

Contoh ini memanggil `DescribeKey` operasi pada kunci KMS asimetris yang digunakan untuk penandatanganan dan verifikasi. Responsnya mencakup algoritma penandatanganan yang AWS KMS mendukung kunci KMS ini.

```
$ 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: Dapatkan kebijakan kunci yang dilampirkan ke kunci KMS

GetKeyPolicyOperasi mendapatkan kebijakan kunci yang dilampirkan ke kunci KMS. Untuk mengidentifikasi kunci KMS, gunakan ID kunci atau kunci ARN. Anda juga harus menentukan nama kebijakan, yang selalu `default`. (Jika output Anda sulit dibaca, tambahkan `--output text` opsi ke perintah Anda.) `GetKeyPolicy`hanya berfungsi pada kunci KMS di akun penelepon dan Wilayah.

Untuk contoh dalam beberapa bahasa pemrograman, lihat Mendapatkan kebijakan kunci.

```
$ 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: Dapatkan nama alias dan ARN untuk kunci KMS

ListAliasesOperasi mengembalikan alias di akun dan Wilayah. `TargetKeyId`Dalam respon menampilkan ID kunci dari kunci KMS yang alias mengacu pada, jika ada.

Secara default, perintah ListAliases mengembalikan semua alias di akun dan wilayah. Ini termasuk alias yang Anda buat dan kaitkan dengan kunci yang dikelola pelanggan Anda, dan alias yang AWS dibuat dan dikaitkan dengan Kunci yang dikelola AWSdi akun Anda. Anda dapat mengenali alias AWS karena nama mereka memiliki format `aws/<service-name>`, seperti `aws/dynamodb`.

Responsnya mungkin juga menyertakan alias tanpa `TargetKeyId` bidang, seperti `aws/redshift` alias dalam contoh ini. Ini adalah alias yang AWS telah ditentukan sebelumnya yang telah dibuat tetapi belum dikaitkan dengan kunci KMS.

Untuk contoh dalam beberapa bahasa pemrograman, lihat Membuat daftar alias.

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


Untuk mendapatkan alias yang merujuk ke kunci KMS tertentu, gunakan parameter. `KeyId` Nilai parameter dapat menjadi ID kunci atau ARN kunci. Anda tidak dapat menentukan nama alias atau ARN alias.

Perintah dalam contoh berikut mendapatkan alias yang merujuk ke kunci yang dikelola pelanggan. Tetapi Anda dapat menggunakan perintah seperti ini untuk menemukan alias yang merujuk Kunci yang dikelola AWSjuga.

```
$ 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
        },
    ]
}
```

Untuk mendapatkan hanya aliasKunci yang dikelola AWS, gunakan fitur bahasa pemrograman Anda untuk memfilter respons.

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

ListResourceTags: Dapatkan tag pada tombol KMS

ListResourceTagsOperasi mengembalikan tag pada kunci KMS yang ditentukan. API mengembalikan tag untuk satu kunci KMS, tetapi Anda dapat menjalankan perintah dalam satu lingkaran untuk mendapatkan tag untuk semua kunci KMS di akun dan Wilayah, atau untuk satu set kunci KMS yang Anda pilih. API ini mengembalikan satu halaman pada satu waktu, jadi jika Anda memiliki banyak tag pada banyak kunci KMS, Anda mungkin harus menggunakan paginator dalam bahasa pemrograman Anda untuk mendapatkan semua tag yang Anda inginkan.

`ListResourceTags`Operasi mengembalikan tag untuk semua kunci KMS, tetapi tidak Kunci yang dikelola AWSditandai. Ini hanya berfungsi pada kunci KMS di akun penelepon dan Wilayah.

Untuk menemukan tag untuk kunci KMS, gunakan `ListResourceTags` operasi. Parameter `KeyId` diperlukan. Ini menerima ID kunci atau ARN kunci. Sebelum menjalankan contoh ini, ganti contoh kunci ARN dengan yang valid.

```
$  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
}
```

Anda mungkin ingin menggunakan `ListResourceTags` operasi untuk mendapatkan semua kunci KMS di akun dan Wilayah dengan tag, kunci tag, atau nilai tag tertentu. Untuk melakukannya, gunakan fitur penyaringan bahasa pemrograman Anda.

Misalnya, skrip Bash berikut menggunakan ListKeysdan `ListResourceTags` operasi untuk mendapatkan semua kunci KMS di akun dan Wilayah dengan kunci `Project` tag. Kedua operasi ini hanya mendapatkan halaman pertama dari hasil. Jika Anda memiliki banyak kunci KMS atau banyak tag, gunakan fitur pagination bahasa Anda untuk mendapatkan seluruh hasil dari setiap operasi. Sebelum menjalankan contoh ini, ganti contoh ID kunci dengan yang valid.

```
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
```

Output diformat seperti contoh output berikut.

```
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"
    }
]
```

Melihat konfigurasi kriptografi kunci KMS

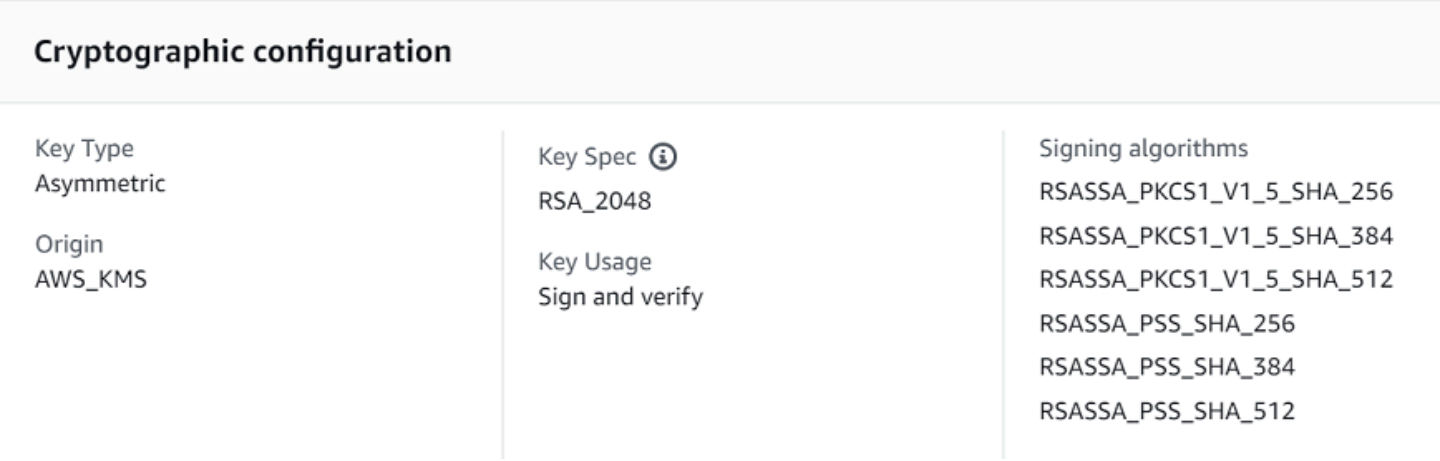
Setelah Anda membuat kunci KMS Anda, Anda dapat melihat konfigurasi kriptografinya. Anda tidak dapat mengubah konfigurasi kunci KMS setelah dibuat. Jika Anda lebih suka konfigurasi yang berbeda, hapus kunci KMS dan buat lagi.

Anda dapat menemukan konfigurasi kriptografi kunci KMS Anda, termasuk spesifikasi kunci, penggunaan kunci, dan enkripsi yang didukung atau algoritma penandatanganan, di AWS KMS konsol atau dengan menggunakan API. AWS KMS Untuk detailnya, lihat Mengidentifikasi kunci KMS asimetris.

Di AWS KMS konsol, halaman detail untuk setiap kunci KMS menyertakan tab konfigurasi Kriptografi yang menampilkan detail kriptografi tentang kunci KMS Anda. Misalnya, gambar berikut menunjukkan tab konfigurasi Kriptografi untuk kunci KMS RSA yang digunakan untuk penandatanganan dan verifikasi.

Tab konfigurasi kriptografi untuk beberapa kunci KMS tujuan khusus memiliki bagian khusus tambahan. Misalnya, tab konfigurasi Kriptografi untuk kunci KMS di toko kunci khusus memiliki

bagian toko kunci Kustom. Tab konfigurasi kriptografi untuk kunci KMS di toko kunci eksternal memiliki bagian kunci Eksternal.



Di AWS KMS API, gunakan DescribeKeyoperasi. `KeyMetadata`Struktur dalam respons mencakup konfigurasi kriptografi kunci KMS. Misalnya, `DescribeKey` mengembalikan respons berikut untuk kunci KMS RSA yang digunakan untuk penandatanganan dan verifikasi.

```
{
  "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"
```

```
    ]
  }
}
```

Menemukan ID kunci dan kunci ARN

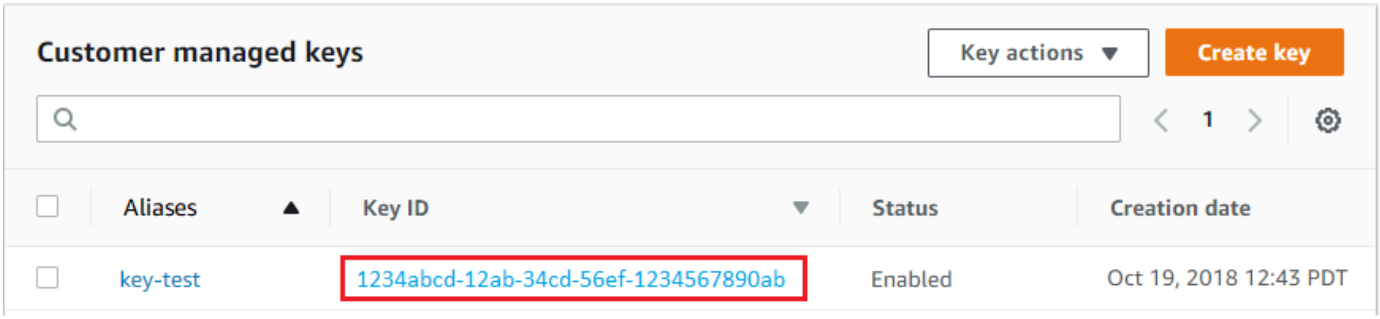
Untuk mengidentifikasiAWS KMS key, Anda dapat menggunakan ID kunci atau Nama Sumber Daya Amazon (kunci ARN). Dalam Operasi kriptografi, Anda juga dapat menggunakan nama alias atau ARN alias.

Untuk informasi rinci tentang pengidentifikasi kunci KMS yang didukung olehAWS KMS, lihat. Pengidentifikasi kunci () KeyId Untuk bantuan menemukan nama alias dan alias ARN, lihat Menemukan nama alias dan ARN alias.

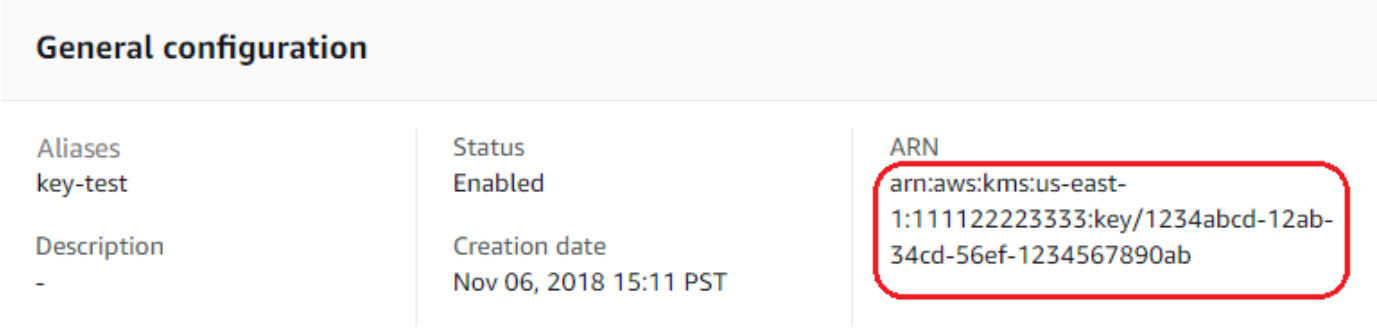
Untuk menemukan ID dan ARN kunci (konsol)

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih kunci yang dikelola AWS.
4. Untuk menemukan ID kunci untuk kunci KMS, lihat baris yang dimulai dengan alias kunci KMS.

   Kolom ID Kunci muncul dalam tabel secara default. Jika kolom ID Kunci tidak muncul di tabel Anda, gunakan prosedur yang dijelaskan di the section called “Menyesuaikan tabel kunci KMS Anda” untuk memulihkannya. Anda juga dapat melihat ID kunci kunci KMS pada halaman detailnya.



5. Untuk menemukan Nama Sumber Daya Amazon (ARN) dari kunci KMS, pilih ID kunci atau alias. ARN kunci muncul di bagian Konfigurasi umum.



Untuk menemukan ID dan ARN kunci (API AWS KMS)

Untuk menemukan ID kunci dan kunci ARN dari sebuahAWS KMS key, gunakan operasi. ListKeys Untuk contoh dalam beberapa bahasa pemrograman, lihat Mendapatkan ID dan ARN kunci dan Dapatkan ID dan ARN kunci.

`ListKeys`Tanggapan tersebut mencakup ID kunci dan kunci ARN untuk setiap kunci KMS di akun dan Wilayah.

```
$ 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"
        }
    ]
}
```

Menemukan nama alias dan ARN alias

Alias adalah nama ramah untuk AWS KMS AWS KMS keys(kunci KMS). Anda dapat menemukan nama alias dan ARN alias di konsol AWS KMS atau API AWS KMS.

Untuk informasi rinci tentang pengidentifikasi kunci KMS yang AWS KMS mendukung, lihat. Pengidentifikasi kunci () KeyId Untuk bantuan menemukan ID kunci dan ARN kunci, lihat Menemukan ID kunci dan kunci ARN.

Topik

- Untuk menemukan nama alias dan ARN alias (konsol)
- Untuk menemukan nama alias dan ARN alias (API AWS KMS)

Untuk menemukan nama alias dan ARN alias (konsol)

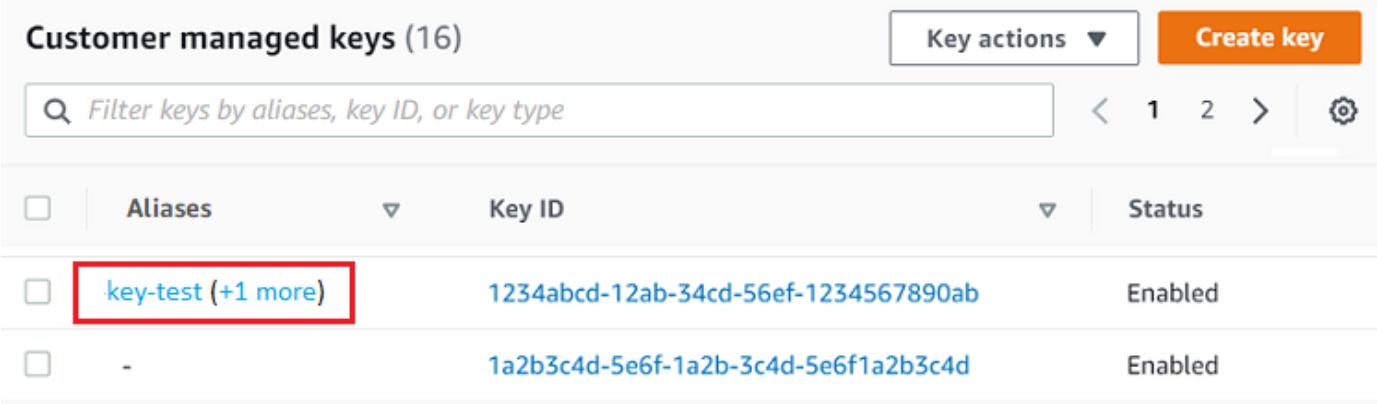
AWS KMSKonsol menampilkan alias yang terkait dengan kunci KMS.

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih kunci yang dikelola AWS.
4. Kolom Alias menampilkan alias untuk setiap tombol KMS. Jika kunci KMS tidak memiliki alias, tanda hubung (-) muncul di kolom Alias.

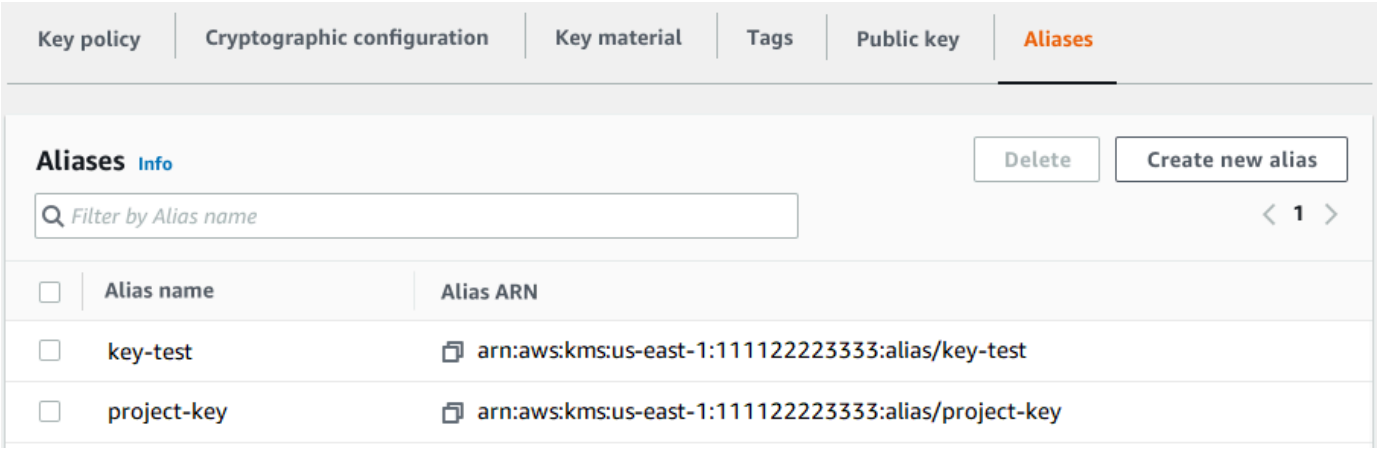
   Jika kunci KMS memiliki beberapa alias, kolom Alias juga memiliki ringkasan alias, seperti (+ n lebih). Misalnya, kunci KMS berikut memiliki dua alias, salah satunya adalah. `key-test`

   Untuk menemukan nama alias dan alias ARN dari semua alias untuk kunci KMS, gunakan tab Alias.

   - Untuk pergi langsung ke tab Alias, di kolom Alias, pilih ringkasan alias (+n lebih banyak). Ringkasan alias hanya muncul jika kunci KMS memiliki lebih dari satu alias.
   - Atau, pilih alias atau ID kunci dari kunci KMS (yang membuka halaman detail untuk tombol KMS) dan kemudian pilih tab Alias. Tab ada di bawah bagian Konfigurasi umum.



5. Tab Alias menampilkan nama alias dan alias ARN dari semua alias untuk kunci KMS. Anda juga dapat membuat dan menghapus alias untuk tombol KMS pada tab ini.



Untuk menemukan nama alias dan ARN alias (API AWS KMS)

Untuk menemukan nama alias dan alias ARN dari sebuahAWS KMS key, gunakan operasi. ListAliases Untuk contoh dalam beberapa bahasa pemrograman, lihat Membuat daftar alias dan Dapatkan nama alias dan ARN.

Secara default, respons termasuk nama alias dan ARN alias untuk setiap alias di akun dan Wilayah. Untuk mendapatkan hanya alias untuk kunci KMS tertentu, gunakan parameter. `KeyId`

Misalnya, perintah berikut hanya mendapatkan alias untuk kunci KMS contoh dengan ID kunci. `1234abcd-12ab-34cd-56ef-1234567890ab`

```
$ 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
        }
    ]
}
```

Mengedit kunci

Anda dapat mengubah properti berikut dari kunci terkelola pelanggan Anda di AWS KMS konsol dan dengan menggunakan AWS KMS API.

Anda tidak dapat mengedit properti apa pun dari Kunci yang dikelola AWSatau Kunci milik AWS. Kunci-kunci ini dikelola oleh AWS layanan yang membuatnya.

Deskripsi

Anda dapat mengubah deskripsi kunci yang dikelola pelanggan Anda di halaman detail untuk kunci KMS atau dengan menggunakan UpdateKeyDescriptionoperasi.

Untuk mengedit deskripsi kunci di konsol, di sudut kanan atas halaman detail untuk tombol KMS, pilih Edit.

Kebijakan kunci

Anda dapat mengubah kebijakan kunci pada tab Kebijakan kunci pada halaman detail untuk kunci yang dikelola pelanggan atau dengan menggunakan PutKeyPolicyoperasi.

Untuk rincian selengkapnya, lihat Mengubah kebijakan kunci.

Tag

Anda dapat membuat dan menghapus tag di halaman kunci terkelola Pelanggan di AWS KMS konsol, atau pada tab Tag di halaman detail untuk kunci yang dikelola pelanggan. Atau Anda dapat menggunakan TagResourcedan UntagResourceoperasi.

Untuk rincian selengkapnya, lihat Tombol penandaan.

Mengaktifkan dan menonaktifkan

Anda dapat mengaktifkan dan menonaktifkan kunci KMS di halaman kunci terkelola Pelanggan di AWS KMS konsol, atau pada halaman detail untuk kunci yang dikelola pelanggan. Atau Anda dapat menggunakan EnableKeydan DisableKeyoperasi.

Untuk rincian selengkapnya, lihat Mengaktifkan dan menonaktifkan kunci.

Rotasi kunci otomatis

Anda dapat mengaktifkan dan menonaktifkan rotasi tombol otomatis pada tab Rotasi kunci pada halaman detail untuk kunci yang dikelola pelanggan atau dengan menggunakan EnableKeyRotationdan DisableKeyRotationoperasi.

Untuk rincian selengkapnya, lihat Berputar AWS KMS keys.

Lihat juga

Memperbarui alias

Tombol penandaan

DiAWS KMS, Anda dapat menambahkan tag ke kunci terkelola pelanggan saat Anda membuat kunci KMS, dan menandai atau menghapus tag kunci KMS yang ada kecuali mereka menunggu penghapusan. Anda tidak dapat menandai alias, toko kunci khusus, Kunci yang dikelola AWSKunci milik AWS, atau kunci KMS di lainnya. Akun AWS Tanda adalah opsional, tetapi tanda bisa sangat berguna.

Untuk informasi lebih lanjut, lihat Membuat kunci dan Mengedit kunci. Untuk informasi umum tentang tag, termasuk praktik terbaik, strategi penandaan, serta format dan sintaks tag, lihat Menandai AWS sumber daya di. Referensi Umum Amazon Web

Topik

- Tentang tanda di AWS KMS

- Mengelola tag kunci KMS di konsol
- Mengelola tag kunci KMS dengan operasi API
- Pengontrolan akses ke tanda
- Menggunakan tag untuk mengontrol akses ke tombol KMS

Tentang tanda di AWS KMS

Tanda adalah label metadata opsional yang dapat Anda tugaskan (atau ditugaskan AWS) ke sumber daya AWS. Setiap tanda terdiri dari kunci tanda dan nilai tanda, keduanya adalah string peka huruf besar/kecil. Nilai tanda bisa berupa string kosong (null). Setiap tanda pada sumber daya harus memiliki kunci tanda yang berbeda, tetapi Anda dapat menambahkan tanda yang sama untuk beberapa sumber daya AWS. Setiap sumber daya dapat memiliki hingga 50 tanda yang dibuat pengguna.

Jangan sertakan informasi rahasia atau sensitif dalam kunci tag atau nilai tag. Tag dapat diakses oleh banyak orangLayanan AWS, termasuk penagihan.

DiAWS KMS, Anda dapat menambahkan tag ke kunci terkelola pelanggan saat Anda membuat kunci KMS, dan menandai atau menghapus tag kunci KMS yang ada kecuali mereka menunggu penghapusan. Anda tidak dapat menandai alias, toko kunci khusus, Kunci yang dikelola AWSKunci milik AWS, atau kunci KMS di lainnya. Akun AWS Tanda adalah opsional, tetapi tanda bisa sangat berguna.

Misalnya, Anda dapat menambahkan `"Project"="Alpha"` tag ke semua kunci KMS dan bucket Amazon S3 yang Anda gunakan untuk proyek Alpha.

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

Untuk informasi umum tentang tag, termasuk format dan sintaks, lihat Menandai AWS sumber daya di. Referensi Umum Amazon Web

Tanda membantu Anda melakukan hal berikut:

- Mengidentifikasi dan mengorganisasi sumber daya AWS Anda. Banyak layanan AWS yang mendukung pemberian tag, sehingga Anda dapat menetapkan tanda yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Misalnya, Anda dapat menetapkan tag yang sama ke kunci KMS dan volume atau rahasia Amazon Elastic

Block Store (Amazon EBS) EBS). AWS Secrets Manager Anda juga dapat menggunakan tag untuk mengidentifikasi kunci KMS untuk otomatisasi.

- Telusuri biaya AWS Anda. Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda Anda. Anda dapat menggunakan fitur ini untuk melacak biaya AWS KMS untuk proyek, aplikasi, atau pusat biaya.

  Untuk informasi selengkapnya tentang penggunaan tanda untuk alokasi biaya, lihat Menggunakan Tanda Alokasi Biaya dalam Panduan Pengguna AWS Billing. Untuk informasi tentang aturan untuk kunci tanda dan nilai tanda, lihat Pembatasan Tanda Ditetapkan Pengguna di AWS BillingPanduan Pengguna.
- Mengendalikan akses ke sumber daya AWS Anda. Mengizinkan dan menolak akses ke kunci KMS berdasarkan tag mereka adalah bagian dari AWS KMS dukungan untuk kontrol akses berbasis atribut (ABAC). Untuk informasi tentang mengontrol akses AWS KMS keys berdasarkan tag mereka, lihatMenggunakan tag untuk mengontrol akses ke tombol KMS. Untuk informasi selengkapnya tentang menggunakan tanda untuk mengontrol akses ke sumber daya AWS, lihat Mengontrol Akses ke sumber daya AWS di Panduan Pengguna IAM.

AWS KMSmenulis entri ke AWS CloudTrail log Anda saat Anda menggunakan TagResource, UntagResource, atau ListResourceTagsoperasi.

Mengelola tag kunci KMS di konsol

Anda dapat menambahkan tag ke kunci KMS saat Anda membuat kunci KMS di konsol. AWS KMS Anda juga dapat menggunakan tab Tag di konsol untuk menambahkan, mengedit, dan menghapus tag pada kunci yang dikelola pelanggan Untuk menambah, mengedit, melihat, dan menghapus tag untuk kunci KMS, Anda harus memiliki izin yang diperlukan. Untuk detailnya, lihat Pengontrolan akses ke tanda.

Tambahkan tag saat membuat kunci KMS

Untuk menambahkan tag saat membuat kunci KMS di konsol, Anda harus memiliki `kms:TagResource` izin dalam kebijakan IAM selain izin yang diperlukan untuk membuat kunci KMS dan melihat kunci KMS di konsol. Minimal, izin harus mencakup semua kunci KMS di akun dan Wilayah.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.

2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. (Anda tidak dapat mengelola tag dariKunci yang dikelola AWS)
4. Pilih jenis kunci, lalu pilih Selanjutnya.
5. Masukkan deskripsi alias dan opsional.
6. Masukkan kunci tanda dan, sebagai pilihan, nilai tanda. Untuk menambahkan tanda tambahan, pilih Tambahkan tanda. Untuk menghapus sebuah tanda, pilih Hapus. Setelah selesai menandai kunci KMS baru, pilih Berikutnya.
7. Selesai membuat kunci KMS Anda.

Melihat dan mengelola tag pada kunci KMS yang ada

Untuk menambahkan, melihat, mengedit, dan menghapus tag di konsol, Anda memerlukan izin penandaan pada kunci KMS. Anda bisa mendapatkan izin ini dari kebijakan kunci untuk kunci KMS atau, jika kebijakan kunci mengizinkannya, dari kebijakan IAM yang menyertakan kunci KMS. Anda memerlukan izin ini selain izin untuk melihat kunci KMS di konsol.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. (Anda tidak dapat mengelola tag dariKunci yang dikelola AWS)
4. Anda dapat menggunakan filter tabel untuk menampilkan hanya tombol KMS dengan tag tertentu. Untuk detailnya, lihat Menyortir dan memfilter kunci KMS Anda.
5. Pilih kotak centang di sebelah alias tombol KMS.
6. Pilih Tindakan kunci, Menambah atau mengedit tanda.
7. Pada halaman detail untuk kunci KMS, pilih tab Tag.

   - Untuk membuat tanda pertama Anda, pilih Buat tanda, ketik kunci tanda (diperlukan) dan nilai tag (opsional), kemudian pilih Simpan.

     Jika Anda membiarkan nilai tanda kosong, nilai tanda yang sebenarnya adalah string null atau kosong.
   - Untuk menambahkan tanda, pilih Edit, pilih Tambahkan tanda, ketik kunci tanda dan nilai tanda, kemudian pilih Simpan.

- Untuk mengubah nama atau nilai tanda, pilih Edit, buat perubahan, lalu pilih Simpan.
- Untuk menghapus sebuah tanda, pilih Edit. Pada baris tanda, pilih Hapus, lalu pilih Simpan.

8. Untuk menyimpan perubahan Anda, memilih Simpan perubahan.

Mengelola tag kunci KMS dengan operasi API

Anda dapat menggunakan AWS Key Management Service(AWS KMS) API untuk menambahkan, menghapus, dan mencantumkan tag untuk kunci KMS yang Anda kelola. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung. Anda tidak dapat menandaiKunci yang dikelola AWS.

Untuk menambahkan, mengedit, melihat, dan menghapus tag untuk kunci KMS, Anda harus memiliki izin yang diperlukan. Untuk detailnya, lihat Pengontrolan akses ke tanda.

Topik

- CreateKey: Tambahkan tag ke kunci KMS baru
- TagResource: Menambahkan atau mengubah tag untuk kunci KMS
- ListResourceTags: Dapatkan tag untuk kunci KMS
- UntagResource: Hapus tag dari kunci KMS

CreateKey: Tambahkan tag ke kunci KMS baru

Anda dapat menambahkan tag saat Anda membuat kunci yang dikelola pelanggan Untuk menentukan tag, gunakan `Tags` parameter CreateKeyoperasi.

Untuk menambahkan tag saat membuat kunci KMS, pemanggil harus memiliki `kms:TagResource` izin dalam kebijakan IAM. Minimal, izin harus mencakup semua kunci KMS di akun dan Wilayah. Untuk rincian selengkapnya, lihat Pengontrolan akses ke tanda.

Nilai dari parameter `Tags` dari `CreateKey` adalah kumpulan kunci tanda peka huruf besar/kecil dan pasangan nilai kunci. Setiap tag pada kunci KMS harus memiliki nama tag yang berbeda. Nilai tanda bisa berupa string kosong atau null.

Misalnya, AWS CLI perintah berikut membuat kunci KMS enkripsi simetris dengan `Project:Alpha` tag. Saat menentukan lebih dari satu pasangan nilai kunci, gunakan spasi untuk memisahkan setiap pasangan.

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

Ketika perintah ini berhasil, ia mengembalikan `KeyMetadata` objek dengan informasi tentang kunci KMS baru. Namun, `KeyMetadata` tidak termasuk tanda. Untuk mendapatkan tag, gunakan ListResourceTagsoperasi.

TagResource: Menambahkan atau mengubah tag untuk kunci KMS

TagResourceOperasi menambahkan satu atau lebih tag ke kunci KMS. Anda tidak dapat menggunakan operasi ini untuk menambah atau mengedit tanda dalam Akun AWS berbeda.

Untuk menambahkan tanda, tentukan kunci tanda baru dan nilai tanda. Untuk mengedit tanda, tentukan kunci tanda yang sudah ada dan nilai tanda baru. Setiap tag pada kunci KMS harus memiliki kunci tag yang berbeda. Nilai tanda bisa berupa string kosong atau null.

Misalnya, perintah berikut menambahkan **`Purpose`** dan **`Department`** tag ke kunci KMS contoh.

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

Ketika perintah ini berhasil, maka tidak mengembalikan output apa pun. Untuk melihat tag pada kunci KMS, gunakan ListResourceTagsoperasi.

Anda juga dapat menggunakan TagResource untuk mengubah nilai tanda dari tanda yang ada. Untuk mengganti nilai tanda, tentukan kunci tanda yang sama dengan nilai yang berbeda.

Sebagai contoh, perintah ini mengubah nilai tanda `Purpose` dari `Pretest` ke `Test`.

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

ListResourceTags: Dapatkan tag untuk kunci KMS

ListResourceTagsOperasi mendapatkan tag untuk kunci KMS. parameter `KeyId` diperlukan. Anda tidak dapat menggunakan operasi ini untuk melihat tag pada tombol KMS secara berbedaAkun AWS.

Misalnya, perintah berikut mendapatkan tag untuk kunci KMS contoh.

```
$ 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: Hapus tag dari kunci KMS

UntagResourceOperasi menghapus tag dari kunci KMS. Untuk mengidentifikasi tanda yang akan dihapus, tentukan kunci tanda. Anda tidak dapat menggunakan operasi ini untuk menghapus tag dari kunci KMS yang berbedaAkun AWS.

Ketika berhasil, operasi `UntagResource` tidak mengembalikan output apa pun. Juga, jika kunci tag yang ditentukan tidak ditemukan pada kunci KMS, itu tidak membuang pengecualian atau mengembalikan respons. Untuk mengonfirmasi bahwa operasi berhasil, gunakan ListResourceTagsoperasi.

Misalnya, perintah ini menghapus **`Purpose`** tag dan nilainya dari kunci KMS yang ditentukan.

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

Pengontrolan akses ke tanda

Untuk menambah, melihat, dan menghapus tanda, baik dalam konsol AWS KMS atau dengan menggunakan API, prinsipal membutuhkan izin penandaan. Anda dapat memberikan izin ini di kebijakan kunci. Anda juga dapat memberikannya dalam kebijakan IAM (termasuk kebijakan VPC endpoint), tetapi hanya jika kebijakan kunci mengizinkannya. Kebijakan

AWSKeyManagementServicePowerUserterkelola memungkinkan prinsipal untuk menandai, menghapus tag, dan mencantumkan tag pada semua kunci KMS yang dapat diakses akun.

Anda juga dapat membatasi izin ini dengan menggunakan kunci kondisi global AWS untuk tanda. DalamAWS KMS, kondisi ini dapat mengontrol akses ke operasi penandaan, seperti TagResourcedan UntagResource.

> **Note**
>
> Berhati-hatilah saat memberikan izin prinsipal untuk mengelola tanda dan alias. Mengubah tag atau alias dapat mengizinkan atau menolak izin ke kunci yang dikelola pelanggan. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.

Untuk kebijakan contoh dan informasi lebih lanjut, lihat Mengontrol Akses Berdasarkan Kunci Tanda di Panduan Pengguna IAM.

Izin untuk membuat dan mengelola tanda bekerja sebagai berikut.

km: TagResource

Memungkinkan prinsipal untuk menambah atau mengedit tanda. Untuk menambahkan tag saat membuat kunci KMS, prinsipal harus memiliki izin dalam kebijakan IAM yang tidak terbatas pada kunci KMS tertentu.

km: ListResourceTags

Memungkinkan prinsipal untuk melihat tag pada tombol KMS.

km: UntagResource

Memungkinkan prinsipal untuk menghapus tag dari kunci KMS.

Izin tanda dalam kebijakan

Anda dapat memberikan izin penandaan dalam kebijakan kunci atau kebijakan IAM. Misalnya, kebijakan kunci contoh berikut memberikan izin penandaan pengguna tertentu pada kunci KMS. Ini memberikan semua pengguna yang dapat mengasumsikan contoh peran Administrator atau Developer izin untuk melihat tanda.

```
{
```

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

Untuk memberikan izin penandaan prinsipal pada beberapa kunci KMS, Anda dapat menggunakan kebijakan IAM. Agar kebijakan ini efektif, kebijakan kunci untuk setiap kunci KMS harus mengizinkan akun menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal untuk membuat kunci KMS. Ini juga memungkinkan mereka untuk membuat dan mengelola tag pada semua kunci KMS di akun

yang ditentukan. Kombinasi ini memungkinkan prinsipal untuk menggunakan parameter Tag CreateKeyoperasi untuk menambahkan tag ke kunci KMS saat mereka membuatnya.

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

Membatasi izin tanda

Anda dapat membatasi izin penandaan dengan menggunakan ketentuan kebijakan. Kondisi kebijakan berikut dapat diterapkan ke izin `kms:TagResource` dan `kms:UntagResource`. Misalnya, Anda dapat menggunakan kondisi `aws:RequestTag/`*`tag-key`* mengizinkan prinsipal untuk menambahkan hanya tanda tertentu, atau mencegah prinsipal menambahkan tanda dengan kunci tanda tertentu. Atau, Anda dapat menggunakan `kms:KeyOrigin` kondisi ini untuk mencegah prinsipal menandai atau melepas tag kunci KMS dengan bahan kunci yang diimpor.

- aws: RequestTag
- aws:ResourceTag/tag-key (hanya kebijakan IAM)
- aws: TagKeys
- km: CallerAccount
- km: KeySpec
- km: KeyUsage

- km: KeyOrigin
- km: ViaService

Sebagai praktik terbaik saat Anda menggunakan tag untuk mengontrol akses ke kunci KMS, gunakan tombol `aws:RequestTag/`*`tag-key`* atau `aws:TagKeys` kondisi untuk menentukan tag (atau kunci tag) mana yang diizinkan.

Sebagai contoh, kebijakan IAM berikut ini mirip dengan yang sebelumnya. Namun, kebijakan ini memungkinkan prinsipal untuk membuat tanda (`TagResource`) dan menghapus tanda `UntagResource` hanya untuk tanda dengan kunci tanda `Project`.

Karena `TagResource` dan `UntagResource` permintaan dapat menyertakan beberapa tag, Anda harus menentukan operator `ForAllValues` atau `ForAnyValue` set dengan TagKeys kondisi aws:. Operator `ForAnyValue` mensyaratkan bahwa setidaknya salah satu kunci tanda dalam permintaan cocok dengan salah satu kunci tanda dalam kebijakan. Operator `ForAllValues` mensyaratkan bahwa semua kunci tanda dalam permintaan cocok dengan salah satu kunci tanda dalam kebijakan. `ForAllValues`Operator juga kembali `true` jika tidak ada tag dalam permintaan, tetapi TagResource dan UntagResource gagal ketika tidak ada tag yang ditentukan. Untuk detail tentang operator set, lihat Gunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

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

Menggunakan tag untuk mengontrol akses ke tombol KMS

Anda dapat mengontrol akses AWS KMS keys berdasarkan tag pada tombol KMS. Misalnya, Anda dapat menulis kebijakan IAM yang memungkinkan prinsipal mengaktifkan dan menonaktifkan hanya kunci KMS yang memiliki tag tertentu. Atau Anda dapat menggunakan kebijakan IAM untuk mencegah prinsipal menggunakan kunci KMS dalam operasi kriptografi kecuali kunci KMS memiliki tag tertentu.

Fitur ini adalah bagian dari dukungan AWS KMS untuk Kontrol akses berbasis atribut (ABAC). Untuk informasi tentang menggunakan tanda untuk mengontrol akses ke sumber daya AWS, lihat Apa guna ABAC untuk AWS? dan Mengontrol Akses ke Sumber Daya AWS Menggunakan Tanda Sumber Daya dalam Panduan Pengguna IAM. Untuk membantu menyelesaikan masalah akses yang terkait dengan ABAC, lihat Memecahkan Masalah ABAC untuk AWS KMS.

> **Note**
>
> Mungkin diperlukan waktu hingga lima menit untuk perubahan tag dan alias untuk memengaruhi otorisasi kunci KMS. Perubahan terbaru mungkin terlihat dalam operasi API sebelum mempengaruhi otorisasi.

AWS KMSmendukung kunci konteks kondisi global aws:ResourceTag/tag-key, yang memungkinkan Anda mengontrol akses ke kunci KMS berdasarkan tag pada kunci KMS. Karena beberapa kunci KMS dapat memiliki tag yang sama, fitur ini memungkinkan Anda menerapkan izin ke satu set kunci KMS tertentu. Anda juga dapat dengan mudah mengubah kunci KMS di set dengan mengubah tag mereka.

Dalam AWS KMS, kunci kondisi `aws:ResourceTag/`*`tag-key`* didukung hanya dalam kebijakan IAM. Ini tidak didukung dalam kebijakan utama, yang hanya berlaku untuk satu kunci KMS, atau pada operasi yang tidak menggunakan kunci KMS tertentu, seperti ListKeysatau ListAliasesoperasi.

Mengontrol akses dengan tanda menyediakan cara sederhana, dapat diskalakan, dan fleksibel untuk mengelola izin. Namun, jika tidak dirancang dan dikelola dengan benar, itu dapat mengizinkan atau menolak akses ke kunci KMS Anda secara tidak sengaja. Jika Anda menggunakan tanda untuk mengontrol akses, pertimbangkan praktik berikut.

- Gunakan tanda untuk memperkuat praktik terbaik dari akses dengan keistimewaan terkecil. Berikan kepala sekolah IAM hanya izin yang mereka butuhkan hanya pada kunci KMS yang harus mereka gunakan atau kelola. Misalnya, gunakan tag untuk memberi label kunci KMS yang digunakan untuk proyek. Kemudian berikan izin tim proyek untuk hanya menggunakan kunci KMS dengan tag proyek.
- Berhati-hatilah tentang memberikan prinsipal izin `kms:TagResource` dan `kms:UntagResource` yang memungkinkan mereka menambahkan, mengedit, dan menghapus tanda. Saat Anda menggunakan tag untuk mengontrol akses ke kunci KMS, mengubah tag dapat memberikan izin kepada prinsipal untuk menggunakan kunci KMS yang tidak diizinkan untuk digunakan. Ini juga dapat menolak akses ke kunci KMS yang diperlukan oleh kepala sekolah lain untuk melakukan pekerjaan mereka. Administrator kunci yang tidak memiliki izin untuk mengubah kebijakan kunci atau membuat hibah dapat mengontrol akses ke kunci KMS jika mereka memiliki izin untuk mengelola tag.

  Jika memungkinkan, gunakan kondisi kebijakan, seperti `aws:RequestTag/`*`tag-key`* atau `aws:TagKeys` untuk membatasi izin penandaan prinsipal untuk tag atau pola tag tertentu pada kunci KMS tertentu.
- Tinjau prinsipal di Akun AWS yang saat ini memiliki izin penandaan dan penghapusan tanda dan menyesuaikannya, jika perlu. Misalnya, kebijakan kunci default konsol untuk administrator kunci menyertakan `kms:TagResource` dan `kms:UntagResource` izin pada kunci KMS tersebut. Kebijakan IAM memungkinkan izin tag dan untag pada semua kunci KMS. Misalnya, kebijakan AWSKeyManagementServicePowerUserterkelola memungkinkan prinsipal untuk menandai, menghapus tag, dan mencantumkan tag pada semua kunci KMS.
- Sebelum menetapkan kebijakan yang bergantung pada tag, tinjau tag pada kunci KMS di tag AndaAkun AWS. Pastikan bahwa kebijakan Anda hanya berlaku untuk tanda yang ingin Anda sertakan. Gunakan CloudTrail log dan CloudWatch alarm untuk mengingatkan Anda untuk menandai perubahan yang mungkin memengaruhi akses ke kunci KMS Anda.
- Kondisi kebijakan berbasis tanda menggunakan pencocokan pola; mereka tidak terikat pada instans tertentu dari tanda. Kebijakan yang menggunakan kunci kondisi berbasis tanda memengaruhi semua tanda baru dan yang sudah ada yang cocok dengan pola. Jika Anda

menghapus dan membuat ulang tanda yang cocok dengan kondisi kebijakan, kondisi berlaku untuk tanda baru, seperti halnya pada tanda lama.

Misalnya, pertimbangkan contoh kebijakan IAM berikut. Hal ini memungkinkan prinsipal untuk memanggil GenerateDataKeyWithoutPlaintextdan mendekripsi operasi hanya pada kunci KMS di akun Anda yang merupakan Wilayah Asia Pasifik (Singapura) dan memiliki tag. `"Project"="Alpha"` Anda mungkin melampirkan kebijakan ini ke peran dalam contoh proyek Alpha.

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

Contoh berikut kebijakan IAM memungkinkan prinsipal untuk menggunakan kunci KMS apa pun di akun untuk operasi kriptografi tertentu. Tapi itu melarang prinsipal menggunakan operasi kriptografi ini pada kunci KMS dengan tag atau tanpa tag. `"Type"="Reserved"` `"Type"`

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

```
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/Type": "true"
        }
      }
    }
  ]
}
```

Mengaktifkan dan menonaktifkan kunci

Anda dapat menonaktifkan dan mengaktifkan kembali kunci yang dikelola pelanggan. Ketika Anda membuat kunci KMS, itu diaktifkan secara default. Jika Anda menonaktifkan kunci KMS, itu tidak dapat digunakan dalam operasi kriptografi apa pun sampai Anda mengaktifkannya kembali.

Karena bersifat sementara dan mudah dibatalkan, menonaktifkan kunci KMS adalah alternatif yang aman untuk menghapus kunci KMS, tindakan yang merusak dan tidak dapat diubah. Jika Anda mempertimbangkan untuk menghapus kunci KMS, nonaktifkan terlebih dahulu dan setel CloudWatch alarm atau mekanisme serupa untuk memastikan bahwa Anda tidak perlu menggunakan kunci untuk mendekripsi data terenkripsi.

Ketika Anda menonaktifkan kunci KMS, itu menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhiLayanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Untuk detailnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Anda tidak dapat mengaktifkan atau menonaktifkan Kunci yang dikelola AWSatau Kunci milik AWS. Kunci yang dikelola AWSdiaktifkan secara permanen untuk digunakan oleh layanan yang menggunakan AWS KMS. Kunci milik AWSdikelola semata-mata oleh layanan yang memilikinya.

> **Note**
>
> AWS KMStidak memutar materi kunci kunci yang dikelola pelanggan saat mereka dinonaktifkan. Untuk informasi selengkapnya, lihat Cara kerja rotasi kunci.

Topik

- Mengaktifkan dan menonaktifkan tombol KMS (konsol)
- Mengaktifkan dan menonaktifkan kunci KMS (API) AWS KMS

Mengaktifkan dan menonaktifkan tombol KMS (konsol)

Anda dapat menggunakan AWS KMS konsol untuk mengaktifkan dan menonaktifkan kunci yang dikelola pelanggan.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih kotak centang untuk tombol KMS yang ingin Anda aktifkan atau nonaktifkan.
5. Untuk mengaktifkan kunci KMS, pilih Tindakan kunci, Aktifkan. Untuk menonaktifkan tombol KMS, pilih Tindakan kunci, Nonaktifkan.

Mengaktifkan dan menonaktifkan kunci KMS (API) AWS KMS

EnableKeyOperasi memungkinkan dinonaktifkanAWS KMS key. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung. Parameter `key-id` diperlukan.

Operasi ini tidak mengembalikan output apa pun. Untuk melihat status kunci, gunakan DescribeKeyoperasi.

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

DisableKeyOperasi menonaktifkan kunci KMS yang diaktifkan. Parameter `key-id` diperlukan.

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

Operasi ini tidak mengembalikan output apa pun. Untuk melihat status kunci, gunakan DescribeKeyoperasi, dan lihat `Enabled` bidangnya.

```
$ 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"
        ]
    }
}
```

Berputar AWS KMS keys

Untuk membuat materi kriptografi baru untuk kunci yang dikelola pelanggan Anda, Anda dapat membuat kunci KMS baru, dan kemudian mengubah aplikasi atau alias Anda untuk menggunakan kunci KMS baru. Atau, Anda dapat memutar materi kunci yang terkait dengan kunci KMS yang ada dengan mengaktifkan rotasi tombol otomatis atau melakukan rotasi sesuai permintaan.

Secara default, ketika Anda mengaktifkan rotasi kunci otomatis untuk kunci KMS, AWS KMS menghasilkan materi kriptografi baru untuk kunci KMS setiap tahun. Anda juga dapat menentukan kustom rotation-period untuk menentukan jumlah hari setelah Anda mengaktifkan rotasi kunci otomatis yang AWS KMS akan memutar materi kunci Anda, dan jumlah hari antara setiap rotasi otomatis sesudahnya. Jika Anda perlu segera memulai rotasi material kunci, Anda dapat melakukan rotasi sesuai permintaan, terlepas dari apakah rotasi kunci otomatis diaktifkan atau tidak. Rotasi sesuai permintaan tidak mengubah jadwal rotasi otomatis yang ada.

AWS KMS menyimpan semua versi sebelumnya dari materi kriptografi selama-lamanya sehingga Anda dapat mendekripsi data apa pun yang dienkripsi dengan kunci KMS itu. AWS KMS tidak menghapus materi kunci yang diputar sampai Anda menghapus kunci KMS. Anda dapat melacak rotasi bahan kunci untuk kunci KMS Anda di Amazon CloudWatch, AWS CloudTrail, dan AWS Key Management Service konsol. Anda juga dapat menggunakan GetKeyRotationStatusoperasi untuk memverifikasi apakah rotasi otomatis diaktifkan untuk kunci KMS dan mengidentifikasi rotasi sesuai permintaan yang sedang berlangsung. Anda dapat menggunakan ListKeyRotationsoperasi untuk melihat rincian rotasi selesai.

Saat Anda menggunakan kunci KMS yang diputar untuk mengenkripsi data, AWS KMS gunakan materi kunci saat ini. Saat Anda menggunakan kunci KMS yang diputar untuk mendekripsi ciphertext, AWS KMS gunakan versi bahan kunci yang digunakan untuk mengenkripsi itu. Anda tidak dapat memilih versi tertentu dari bahan utama untuk operasi dekripsi, AWS KMS secara otomatis memilih

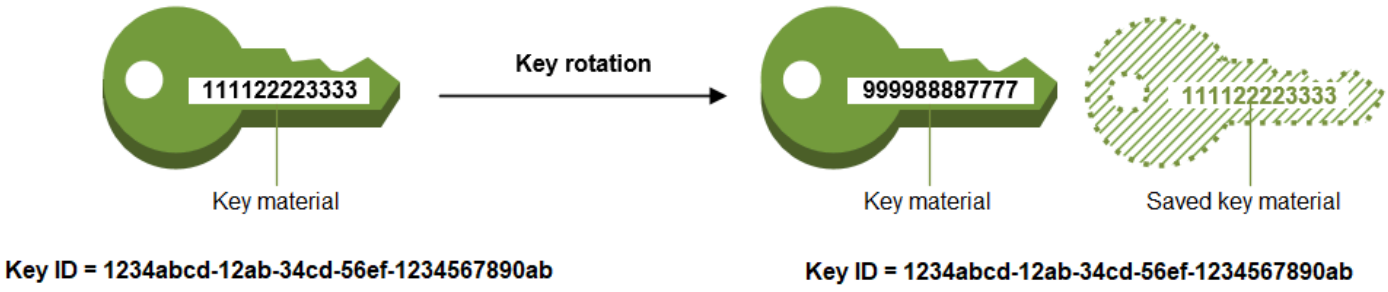
versi yang benar. Karena AWS KMS secara transparan mendekripsi dengan bahan kunci yang sesuai, Anda dapat dengan aman menggunakan kunci KMS yang diputar dalam aplikasi dan tanpa perubahan kode. Layanan AWS

Namun, rotasi tombol otomatis tidak berpengaruh pada data yang dilindungi oleh kunci KMS. Itu tidak memutar kunci data yang dihasilkan oleh kunci KMS atau mengenkripsi ulang data apa pun yang dilindungi oleh kunci KMS, dan itu tidak akan mengurangi efek dari kunci data yang dikompromikan.

AWS KMS mendukung rotasi kunci otomatis dan sesuai permintaan hanya untuk kunci KMS enkripsi simetris dengan bahan kunci yang dibuat. AWS KMS Rotasi otomatis adalah opsional untuk kunci KMS yang dikelola pelanggan. AWS KMS selalu memutar bahan kunci untuk kunci KMS yang AWS dikelola setiap tahun. Rotasi kunci KMS yang AWS dimiliki dikelola oleh AWS layanan yang memiliki kunci.

> **Note**
>
> Periode rotasi untuk Kunci yang dikelola AWS berubah pada Mei 2022. Lihat perinciannya di Kunci yang dikelola AWS.

Rotasi kunci hanya mengubah materi kunci, yang merupakan rahasia kriptografi yang digunakan dalam operasi enkripsi. Kunci KMS adalah sumber daya logis yang sama, terlepas dari apakah atau berapa kali materi utamanya berubah. Properti kunci KMS tidak berubah, seperti yang ditunjukkan pada gambar berikut.



Anda mungkin memutuskan untuk membuat kunci KMS baru dan menggunakannya sebagai pengganti kunci KMS asli. Ini memiliki efek yang sama seperti memutar bahan kunci dalam kunci KMS yang ada, sehingga sering dianggap sebagai memutar kunci secara manual. Rotasi manual adalah pilihan yang baik ketika Anda ingin memutar tombol KMS yang tidak memenuhi syarat untuk rotasi kunci otomatis, termasuk kunci KMS asimetris, kunci KMSHMAC, kunci KMS di toko kuncikhusus, dan kunci KMS dengan bahan kunci impor.

Rotasi kunci dan harga

AWS KMS membebankan biaya bulanan untuk rotasi pertama dan kedua bahan kunci yang dipertahankan untuk kunci KMS Anda. Kenaikan harga ini dibatasi pada rotasi kedua, dan setiap rotasi berikutnya tidak akan ditagih. Untuk detail selengkapnya, lihat Harga AWS Key Management Service.

> **Note**
>
> Anda dapat menggunakan AWS Cost Explorer Serviceuntuk melihat rincian biaya penyimpanan kunci Anda. Misalnya, Anda dapat memfilter tampilan untuk melihat total biaya untuk kunci yang ditagih sebagai kunci KMS saat ini dan yang diputar dengan menentukan **`$REGION-KMS-Keys`** Jenis Penggunaan dan mengelompokkan data berdasarkan Operasi API.
> Anda mungkin masih melihat contoh operasi `Unknown` API lama untuk tanggal historis.

Rotasi kunci dan kuota

Setiap kunci KMS dihitung sebagai satu kunci saat menghitung kuota sumber daya kunci, terlepas dari jumlah versi material kunci yang diputar.

Untuk informasi rinci tentang materi utama dan rotasi, lihat Detail AWS Key Management Service Kriptografi.

Topik

- Mengapa memutar tombol KMS?
- Cara kerja rotasi kunci
- Cara mengaktifkan dan menonaktifkan rotasi kunci otomatis
- Cara melakukan rotasi kunci sesuai permintaan
- Memutar kunci secara manual

Mengapa memutar tombol KMS?

Praktik terbaik kriptografi mencegah penggunaan kembali kunci yang mengenkripsi data secara langsung, seperti kunci data yang dihasilkan. AWS KMS Ketika kunci data 256-bit mengenkripsi

jutaan pesan, mereka dapat menjadi kelelahan dan mulai menghasilkan ciphertext dengan pola halus yang dapat dimanfaatkan oleh aktor pintar untuk menemukan bit dalam kunci. Untuk menghindari kelelahan kunci ini, yang terbaik adalah menggunakan kunci data sekali, atau hanya beberapa kali, yang secara efektif memutar materi kunci.

Namun, kunci KMS paling sering digunakan sebagai kunci pembungkus, juga dikenal sebagai kunci enkripsi kunci. Alih-alih mengenkripsi data, kunci pembungkus mengenkripsi kunci data yang mengenkripsi data Anda. Dengan demikian, mereka digunakan jauh lebih jarang daripada kunci data, dan hampir tidak pernah cukup digunakan kembali untuk risiko kelelahan kunci.

Meskipun risiko kelelahan yang sangat rendah ini, Anda mungkin diminta untuk memutar kunci KMS Anda karena aturan bisnis atau kontrak atau peraturan pemerintah. Ketika Anda dipaksa untuk memutar tombol KMS, kami sarankan Anda menggunakan rotasi tombol otomatis di mana itu didukung, dan rotasi tombol manual ketika rotasi tombol otomatis tidak didukung.

Anda dapat mempertimbangkan untuk melakukan rotasi sesuai permintaan untuk menunjukkan kemampuan rotasi material utama atau untuk memvalidasi skrip otomatisasi. Kami merekomendasikan penggunaan rotasi sesuai permintaan untuk rotasi yang tidak direncanakan, dan menggunakan rotasi kunci otomatis dengan periode rotasi khusus bila memungkinkan.

Cara kerja rotasi kunci

Rotasi kunci dalam AWS KMS dirancang agar transparan dan mudah digunakan. AWS KMS mendukung rotasi kunci otomatis dan sesuai permintaan opsional hanya untuk kunci yang dikelola pelanggan.

Rotasi kunci otomatis

AWS KMS memutar tombol KMS secara otomatis pada tanggal rotasi berikutnya yang ditentukan oleh periode rotasi Anda. Anda tidak perlu mengingat atau menjadwalkan pembaruan.

Rotasi sesuai permintaan

Segera mulai rotasi materi kunci yang terkait dengan kunci KMS Anda, terlepas dari apakah rotasi tombol otomatis diaktifkan atau tidak.

Mengelola materi utama

AWS KMS mempertahankan semua materi kunci untuk kunci KMS, bahkan jika rotasi tombol dinonaktifkan. AWS KMS menghapus materi kunci hanya ketika Anda menghapus kunci KMS.

Menggunakan bahan utama

Saat Anda menggunakan kunci KMS yang diputar untuk mengenkripsi data, AWS KMS gunakan materi kunci saat ini. Saat Anda menggunakan kunci KMS yang diputar untuk mendekripsi ciphertext, AWS KMS gunakan versi yang sama dari bahan kunci yang digunakan untuk mengenkripsi itu. Anda tidak dapat memilih versi tertentu dari bahan utama untuk operasi dekripsi, AWS KMS secara otomatis memilih versi yang benar.

Periode rotasi

Periode rotasi menentukan jumlah hari setelah Anda mengaktifkan rotasi kunci otomatis yang AWS KMS akan memutar materi kunci Anda, dan jumlah hari antara setiap rotasi kunci otomatis sesudahnya. Jika Anda tidak menentukan nilai `RotationPeriodInDays` saat Anda mengaktifkan rotasi kunci otomatis, nilai defaultnya adalah 365 hari.

Anda dapat menggunakan kms: RotationPeriodInDays condition key untuk lebih membatasi nilai-nilai yang prinsipal dapat menentukan dalam parameter. `RotationPeriodInDays`

Tanggal rotasi

AWS KMS secara otomatis memutar tombol KMS pada tanggal rotasi yang ditentukan oleh periode rotasi Anda. Periode rotasi default adalah 365 hari.

Kunci yang dikelola pelanggan

Karena rotasi kunci otomatis bersifat opsional pada kunci yang dikelola pelanggan dan dapat diaktifkan dan dinonaktifkan kapan saja, tanggal rotasi tergantung pada tanggal rotasi terakhir diaktifkan. Tanggal dapat berubah jika Anda mengubah periode rotasi untuk kunci yang sebelumnya Anda aktifkan rotasi tombol otomatis. Tanggal rotasi dapat berubah berkali-kali selama masa pakai kunci.

Misalnya, jika Anda membuat kunci terkelola pelanggan pada 1 Januari 2022, dan mengaktifkan rotasi kunci otomatis dengan periode rotasi default 365 hari pada 15 Maret 2022, AWS KMS putar materi kunci pada 15 Maret 2023, 15 Maret 2024, dan setiap 365 hari setelahnya.

Contoh berikut mengasumsikan bahwa rotasi kunci otomatis diaktifkan dengan periode rotasi default 365 hari. Contoh-contoh ini menunjukkan kasus-kasus khusus yang mungkin memengaruhi periode rotasi kunci.

- Nonaktifkan rotasi tombol - Jika Anda menonaktifkan rotasi tombol otomatis pada titik mana pun, tombol KMS terus menggunakan versi bahan kunci yang digunakannya saat rotasi

dinonaktifkan. Jika Anda mengaktifkan rotasi tombol otomatis lagi, AWS KMS putar materi kunci berdasarkan tanggal pengaktifan rotasi baru.

- Tombol KMS dinonaktifkan - Sementara kunci KMS dinonaktifkan, AWS KMS tidak memutarnya. Namun, status rotasi kunci tidak berubah, dan Anda tidak dapat mengubahnya saat kunci KMS dinonaktifkan. Ketika kunci KMS diaktifkan kembali, jika materi kunci melewati tanggal rotasi terjadwal terakhirnya, AWS KMS putar segera. Jika materi kunci tidak melewatkan tanggal rotasi terakhirnya yang dijadwalkan, AWS KMS lanjutkan jadwal rotasi kunci asli.
- Kunci KMS tertunda penghapusan - Sementara kunci KMS sedang menunggu penghapusan, tidak memutarnya. AWS KMS Status rotasi kunci diatur ke `false` dan Anda tidak dapat mengubahnya saat penghapusan tertunda. Jika penghapusan dibatalkan, status rotasi kunci sebelumnya akan dipulihkan. Jika bahan kunci melewati tanggal rotasi terakhirnya yang dijadwalkan, segera AWS KMS putar. Jika materi kunci tidak melewatkan tanggal rotasi terakhirnya yang dijadwalkan, AWS KMS lanjutkan jadwal rotasi kunci asli.

Kunci yang dikelola AWS

AWS KMS secara otomatis berputar Kunci yang dikelola AWS setiap tahun (sekitar 365 hari). Anda tidak dapat mengaktifkan atau menonaktifkan rotasi kunci untuk Kunci yang dikelola AWS.

Bahan kunci untuk sebuah pertama kali Kunci yang dikelola AWS diputar satu tahun setelah tanggal pembuatannya, dan setiap tahun (sekitar 365 hari dari rotasi terakhir) sesudahnya.

> **Note**
>
> Pada Mei 2022, AWS KMS mengubah jadwal rotasi Kunci yang dikelola AWS dari setiap tiga tahun (sekitar 1.095 hari) menjadi setiap tahun (sekitar 365 hari).
> Baru Kunci yang dikelola AWS secara otomatis diputar satu tahun setelah dibuat, dan kira-kira setiap tahun setelahnya.
> Yang Kunci yang dikelola AWS ada secara otomatis diputar satu tahun setelah rotasi terbaru mereka, dan setiap tahun setelahnya.

Kunci milik AWS

Anda tidak dapat mengaktifkan atau menonaktifkan rotasi kunci untuk Kunci milik AWS. Strategi rotasi kunci untuk sebuah Kunci milik AWS ditentukan oleh AWS layanan yang

membuat dan mengelola kunci. Untuk detail selengkapnya, lihat topik Enkripsi Tidak Aktif di panduan pengguna atau panduan developer untuk layanan tersebut.

Jenis kunci KMS yang didukung

Rotasi kunci otomatis hanya didukung pada kunci KMS enkripsi simetris dengan bahan kunci yang AWS KMS menghasilkan (Asal = AWS_KMS).

Rotasi tombol otomatis tidak didukung pada jenis tombol KMS berikut, tetapi Anda dapat memutar tombol KMS ini secara manual.

- Tombol Asymmetric KMS
- Kunci HMAC KMS
- Kunci KMS di toko kunci khusus
- Kunci KMS dengan bahan kunci impor

Kunci Multi-Wilayah

Anda dapat mengaktifkan dan menonaktifkan rotasi kunci otomatis untuk kunci multi-Wilayah. Anda mengatur properti hanya pada kunci primer. Saat AWS KMS menyinkronkan kunci, ia menyalin pengaturan properti dari kunci utama ke kunci replika. Ketika bahan kunci dari kunci utama diputar, AWS KMS secara otomatis menyalin materi kunci itu ke semua kunci replika. Lihat perinciannya di Memutar kunci multi-Wilayah.

AWS layanan

Anda dapat mengaktifkan rotasi kunci otomatis pada kunci terkelola pelanggan yang Anda gunakan untuk enkripsi sisi server dalam layanan. AWS Rotasi tahunan transparan dan kompatibel dengan layanan AWS .

Memantau rotasi kunci

Saat AWS KMS memutar materi kunci untuk kunci yang dikelola pelanggan Kunci yang dikelola AWSatau pelanggan, ia menulis `KMS CMK Rotation` acara ke Amazon EventBridge dan RotateKey acara ke AWS CloudTrail log Anda. Anda dapat menggunakan catatan ini untuk memverifikasi bahwa kunci KMS telah diputar.

Anda dapat menggunakan AWS Key Management Service konsol untuk melihat jumlah rotasi sesuai permintaan yang tersisa dan daftar semua rotasi material kunci yang diselesaikan untuk kunci KMS.

Anda dapat menggunakan ListKeyRotationsoperasi untuk melihat rincian rotasi selesai.

Konsistensi akhirnya

Rotasi kunci tunduk pada efek konsistensi akhirnya yang sama seperti operasi AWS KMS manajemen lainnya. Mungkin ada sedikit penundaan sebelum materi kunci baru tersedia di seluruh AWS KMS. Namun, memutar materi kunci tidak menyebabkan gangguan atau keterlambatan dalam operasi kriptografi. Materi kunci saat ini digunakan dalam operasi kriptografi sampai bahan kunci baru tersedia di seluruh AWS KMS. Ketika materi kunci untuk kunci Multi-wilayah diputar secara otomatis, AWS KMS gunakan materi kunci saat ini hingga materi kunci baru tersedia di semua Wilayah dengan kunci Multi-wilayah terkait.

Cara mengaktifkan dan menonaktifkan rotasi kunci otomatis

Secara default, ketika Anda mengaktifkan rotasi kunci otomatis untuk kunci KMS, AWS KMS menghasilkan materi kriptografi baru untuk kunci KMS setiap tahun. Anda juga dapat menentukan kustom rotation-period untuk menentukan jumlah hari setelah Anda mengaktifkan rotasi kunci otomatis yang AWS KMS akan memutar materi kunci Anda, dan jumlah hari antara setiap rotasi otomatis sesudahnya.

Rotasi kunci otomatis memiliki manfaat sebagai berikut:

- Properti kunci KMS, termasuk ID kunci, ARN kunci, wilayah, kebijakan, dan izin, tidak berubah ketika kunci diputar.
- Anda tidak perlu mengubah aplikasi atau alias yang merujuk ke ID kunci atau ARN kunci dari kunci KMS.
- Bahan kunci yang berputar tidak mempengaruhi penggunaan kunci KMS di mana pun. Layanan AWS
- Setelah Anda mengaktifkan rotasi kunci, AWS KMS putar tombol KMS secara otomatis pada tanggal rotasi berikutnya yang ditentukan oleh periode rotasi Anda. Anda tidak perlu mengingat atau menjadwalkan pembaruan.

Pengguna yang berwenang dapat menggunakan AWS KMS konsol dan AWS KMS API untuk mengaktifkan dan menonaktifkan rotasi kunci otomatis dan melihat status rotasi kunci.

Topik

- Mengaktifkan dan menonaktifkan rotasi tombol otomatis (konsol)
- Mengaktifkan dan menonaktifkan rotasi kunci otomatis (API)AWS KMS

Mengaktifkan dan menonaktifkan rotasi tombol otomatis (konsol)

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. (Anda tidak dapat mengaktifkan atau menonaktifkan rotasi Kunci yang dikelola AWS. Mereka secara otomatis diputar setiap tahun.)
4. Pilih alias atau ID kunci dari kunci KMS.
5. Pilih tab Rotasi kunci.

   Tab Rotasi kunci hanya muncul di halaman detail kunci KMS enkripsi simetris dengan bahan kunci yang AWS KMS dihasilkan (Asal adalah AWS_KMS), termasuk kunci KMS enkripsi simetris Multi-wilayah.

   Anda tidak dapat secara otomatis memutar kunci KMS asimetris, kunci KMS HMAC, kunci KMS dengan bahan kunci impor, atau kunci KMS di toko kunci khusus. Namun, Anda dapat memutarnya secara manual.
6. Di bagian Rotasi tombol otomatis, pilih Edit.
7. Untuk Rotasi tombol, pilih Aktifkan.

   > **Note**
   >
   > Jika kunci KMS dinonaktifkan atau penghapusan tertunda, AWS KMS tidak memutar materi kunci dan Anda tidak dapat memperbarui status rotasi tombol otomatis atau periode rotasi. Aktifkan tombol KMS atau batalkan penghapusan untuk memperbarui konfigurasi rotasi tombol otomatis. Untuk detailnya, lihat Cara kerja rotasi kunci dan Status AWS KMS kunci kunci.
8. (Opsional) Ketik periode rotasi antara 90 dan 2560 hari. Nilai defaultnya adalah 365 hari. Jika Anda tidak menentukan periode rotasi kustom, AWS KMS akan memutar bahan kunci setiap tahun.

   Anda dapat menggunakan kms: RotationPeriodInDays condition key untuk membatasi nilai yang dapat ditentukan oleh prinsipal untuk periode rotasi.
9. Pilih Simpan.

Mengaktifkan dan menonaktifkan rotasi kunci otomatis (API)AWS KMS

Anda dapat menggunakan AWS Key Management Service (AWS KMS) API untuk mengaktifkan dan menonaktifkan rotasi kunci otomatis, dan melihat status rotasi saat ini dari setiap kunci yang dikelola pelanggan. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

EnableKeyRotationOperasi ini memungkinkan rotasi tombol otomatis untuk kunci KMS yang ditentukan. DisableKeyRotationOperasi menonaktifkannya. Untuk mengidentifikasi kunci KMS dalam operasi ini, gunakan ID kunci atau kunci ARN. Secara default, rotasi kunci dinonaktifkan untuk kunci yang dikelola pelanggan.

Anda dapat menggunakan kms: RotationPeriodInDays condition key untuk membatasi nilai yang dapat ditentukan oleh prinsipal untuk `RotationPeriodInDays` parameter permintaan. `EnableKeyRotation`

Contoh berikut memungkinkan rotasi kunci dengan periode rotasi 180 hari pada kunci KMS enkripsi simetris yang ditentukan dan menggunakan GetKeyRotationStatusoperasi untuk melihat hasilnya. Kemudian, itu menonaktifkan rotasi kunci dan, sekali lagi, menggunakan GetKeyRotationStatus untuk melihat perubahannya.

```
$ 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
}
```

Cara melakukan rotasi kunci sesuai permintaan

Anda dapat melakukan rotasi sesuai permintaan dari materi utama dalam kunci KMS yang dikelola pelanggan, terlepas dari apakah rotasi kunci otomatis diaktifkan atau tidak. Menonaktifkan rotasi otomatis (DisableKeyRotation) tidak memengaruhi kemampuan Anda untuk melakukan rotasi sesuai permintaan, juga tidak membatalkan rotasi sesuai permintaan yang sedang berlangsung. Rotasi sesuai permintaan tidak mengubah jadwal rotasi otomatis yang ada. Misalnya, pertimbangkan kunci KMS yang memiliki rotasi tombol otomatis diaktifkan dengan periode rotasi 730 hari. Jika kunci dijadwalkan untuk berputar secara otomatis pada 14 April 2024, dan Anda melakukan rotasi sesuai permintaan pada 10 April 2024, kunci akan berputar secara otomatis, sesuai jadwal, pada 14 April 2024 dan setiap 730 hari setelahnya.

Anda dapat melakukan rotasi kunci sesuai permintaan maksimal 10 kali per tombol KMS. Anda dapat menggunakan AWS KMS konsol untuk melihat jumlah rotasi sesuai permintaan yang tersisa yang tersedia untuk kunci KMS.

Rotasi kunci sesuai permintaan hanya didukung pada kunci KMS enkripsi simetris. Anda tidak dapat melakukan rotasi sesuai permintaan kunci KMS asimetris, kunci KMS HMAC, kunci KMS dengan bahan kunci impor, atau kunci KMS di toko kunci khusus. Untuk melakukan rotasi sesuai permintaan dari satu set kunci Multi-wilayah terkait, panggil rotasi sesuai permintaan pada kunci utama.

Pengguna yang berwenang dapat menggunakan AWS KMS konsol dan AWS KMS API untuk memulai rotasi kunci sesuai permintaan dan melihat status rotasi kunci.

Topik

- Memulai rotasi kunci sesuai permintaan (konsol)
- Memulai rotasi kunci sesuai permintaan (API)AWS KMS

Memulai rotasi kunci sesuai permintaan (konsol)

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan. (Anda tidak dapat melakukan rotasi sesuai permintaan. Kunci yang dikelola AWS Mereka secara otomatis diputar setiap tahun.)
4. Pilih alias atau ID kunci dari kunci KMS.

5. Pilih tab Rotasi kunci.

   Tab Rotasi kunci hanya muncul di halaman detail kunci KMS enkripsi simetris dengan bahan kunci yang AWS KMS dihasilkan (Asal adalah AWS_KMS), termasuk kunci KMS enkripsi simetris Multi-wilayah.

   Anda tidak dapat melakukan rotasi sesuai permintaan kunci KMS asimetris, kunci KMS HMAC, kunci KMS dengan bahan kunci impor, atau kunci KMS di toko kunci khusus. Namun, Anda dapat memutarnya secara manual.
6. Di bagian rotasi tombol On-Demand, pilih tombol Rotate.
7. Baca dan pertimbangkan peringatan dan informasi tentang jumlah rotasi sesuai permintaan yang tersisa untuk kunci tersebut. Jika Anda memutuskan bahwa Anda tidak ingin melanjutkan rotasi sesuai permintaan, pilih Batalkan.
8. Pilih tombol Putar untuk mengonfirmasi rotasi sesuai permintaan.

> **Note**
>
> Rotasi on-demand tunduk pada efek konsistensi akhirnya yang sama seperti operasi AWS KMS manajemen lainnya. Mungkin ada sedikit penundaan sebelum materi kunci baru tersedia di seluruh AWS KMS. Spanduk di bagian atas konsol memberi tahu Anda saat rotasi sesuai permintaan selesai.

Memulai rotasi kunci sesuai permintaan (API)AWS KMS

Anda dapat menggunakan AWS Key Management Service (AWS KMS) API untuk memulai rotasi kunci sesuai permintaan, dan melihat status rotasi saat ini dari setiap kunci yang dikelola pelanggan. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

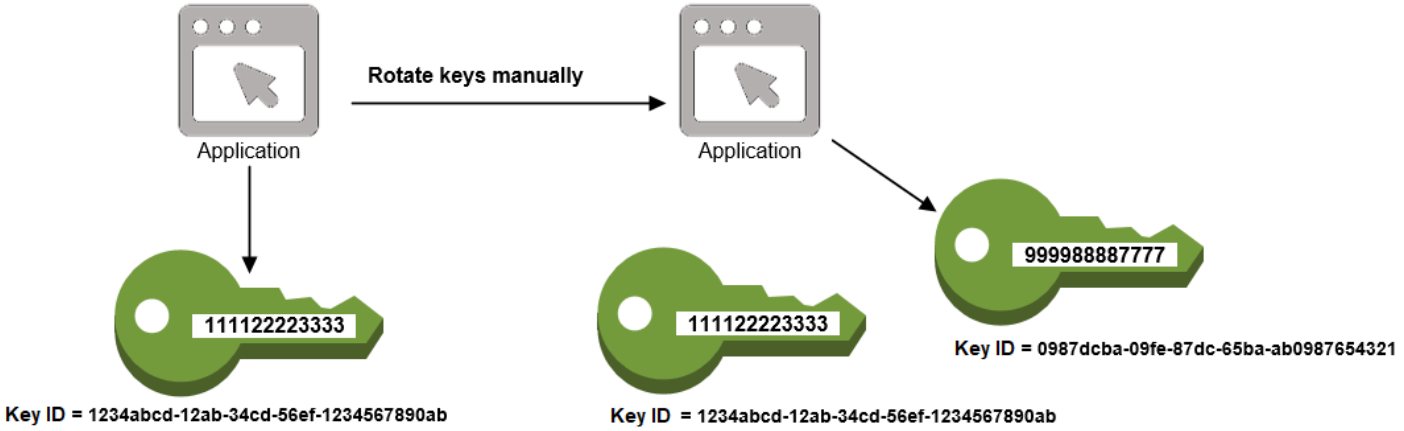
RotateKeyOnDemandOperasi segera memulai rotasi kunci sesuai permintaan untuk kunci KMS yang ditentukan. Untuk mengidentifikasi kunci KMS dalam operasi ini, gunakan ID kunci atau kunci ARN.

Contoh berikut memulai rotasi kunci sesuai permintaan pada kunci KMS enkripsi simetris yang ditentukan dan menggunakan GetKeyRotationStatusoperasi untuk memverifikasi bahwa rotasi sesuai permintaan sedang berlangsung. The `OnDemandRotationStartDate` in the `kms:GetKeyRotationStatus` response mengidentifikasi tanggal dan waktu rotasi on-demand yang sedang berlangsung dimulai.

```
$ 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
}
```

Memutar kunci secara manual

Anda mungkin ingin membuat kunci KMS baru dan menggunakannya sebagai pengganti kunci KMS saat ini alih-alih mengaktifkan rotasi tombol otomatis. Ketika kunci KMS baru memiliki bahan kriptografi yang berbeda dari kunci KMS saat ini, menggunakan kunci KMS baru memiliki efek yang sama seperti mengubah materi kunci dalam kunci KMS yang ada. Proses mengganti satu kunci KMS dengan yang lain dikenal sebagai rotasi kunci manual.



Rotasi manual adalah pilihan yang baik ketika Anda ingin memutar tombol KMS yang tidak memenuhi syarat untuk rotasi kunci otomatis, seperti kunci KMS asimetris, kunci KMS HMAC, kunci KMS di toko kunci khusus, dan kunci KMS dengan bahan kunci impor.

> **Note**
>
> Ketika Anda mulai menggunakan kunci KMS baru, pastikan untuk menjaga kunci KMS asli diaktifkan sehingga AWS KMS dapat mendekripsi data yang kunci KMS asli dienkripsi.

Saat Anda memutar tombol KMS secara manual, Anda juga perlu memperbarui referensi ke ID kunci KMS atau ARN kunci di aplikasi Anda. Alias, yang mengaitkan nama ramah dengan kunci KMS, dapat membuat proses ini lebih mudah. Gunakan alias untuk merujuk ke kunci KMS di aplikasi Anda. Kemudian, ketika Anda ingin mengubah kunci KMS yang digunakan aplikasi, alih-alih mengedit kode aplikasi Anda, ubah kunci KMS target alias. Lihat perinciannya di Menggunakan alias dalam aplikasi Anda.

> **Note**
>
> Alias yang mengarah ke versi terbaru dari kunci KMS yang diputar secara manual adalah solusi yang baik untuk operasi DescribeKey, Enkripsi,,,, GenerateDataKeyGenerateDataKeyPairGenerateMac, dan Tanda Tangan. Alias tidak diizinkan dalam operasi yang mengelola kunci KMS, seperti DisableKeyatau. ScheduleKeyDeletion
>
> Saat memanggil operasi Dekripsi pada kunci KMS enkripsi simetris yang diputar secara manual, hilangkan parameter dari perintah. `KeyId` AWS KMS secara otomatis menggunakan kunci KMS yang mengenkripsi ciphertext.
>
> `KeyId`Parameter diperlukan saat memanggil `Decrypt` atau Verifikasi dengan kunci KMS asimetris, atau memanggil VerifyMacdengan kunci HMAC KMS. Permintaan ini gagal ketika nilai `KeyId` parameter adalah alias yang tidak lagi menunjuk ke kunci KMS yang melakukan operasi kriptografi, seperti ketika kunci diputar secara manual. Untuk menghindari kesalahan ini, Anda harus melacak dan menentukan kunci KMS yang benar untuk setiap operasi.

Untuk mengubah kunci KMS target alias, gunakan UpdateAliasoperasi di API. AWS KMS Misalnya, perintah ini memperbarui `alias/TestKey` alias untuk menunjuk ke kunci KMS baru. Karena operasi tidak mengembalikan output apa pun, contoh menggunakan ListAliasesoperasi untuk menunjukkan bahwa alias sekarang dikaitkan dengan kunci KMS yang berbeda dan `LastUpdatedDate` bidang diperbarui. ListAliases Perintah menggunakan `query`parameter dalam AWS CLI untuk mendapatkan hanya `alias/TestKey` alias.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1521097200.123
        },
    ]
}


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

$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1604958290.722
        },
    ]
}
```

Memantau AWS KMS keys

Pemantauan adalah bagian penting untuk memahami ketersediaan, keadaan, dan penggunaan Anda AWS KMS keys di dalam AWS KMS dan menjaga keandalan, ketersediaan, dan kinerja AWS solusi Anda. Mengumpulkan data pemantauan dari semua bagian solusi AWS akan membantu Anda melakukan debug pada gagal beberapa titik jika ada yang terjadi. Sebelum Anda mulai memantau kunci KMS Anda, buatlah rencana pemantauan yang mencakup jawaban atas pertanyaan-pertanyaan berikut:

- Apa saja sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?

- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu ketika terjadi sesuatu?

Langkah selanjutnya adalah memantau kunci KMS Anda dari waktu ke waktu untuk menetapkan dasar untuk AWS KMS penggunaan normal dan harapan di lingkungan Anda. Saat Anda memantau kunci KMS Anda, simpan data pemantauan historis sehingga Anda dapat membandingkannya dengan data saat ini, mengidentifikasi pola dan anomali normal, dan merancang metode untuk mengatasi masalah.

Misalnya, Anda dapat memantau aktivitas AWS KMS API dan peristiwa yang memengaruhi kunci KMS Anda. Saat data berada di atas atau di bawah norma yang telah ditetapkan, Anda mungkin perlu menyelidiki atau mengambil tindakan korektif.

Untuk menetapkan baseline pola normal, pantau item berikut:

- Aktivitas API AWS KMS untuk operasi bidang data. Ini adalah operasi kriptografi yang menggunakan kunci KMS, seperti Dekripsi, Enkripsi, dan. ReEncryptGenerateDataKey
- Aktivitas API AWS KMS untuk operasi bidang kontrol yang penting bagi Anda. Operasi ini mengelola kunci KMS, dan Anda mungkin ingin memantau mereka yang mengubah ketersediaan kunci KMS (seperti ScheduleKeyDeletion,,,, CancelKeyDeletionDisableKey, EnableKeyImportKeyMaterial, dan DeleteImportedKeyMaterial) atau mengubah kontrol akses kunci KMS (seperti PutKeyPolicydan). RevokeGrant
- AWS KMSMetrik lainnya (seperti jumlah waktu yang tersisa hingga materi kunci impor Anda kedaluwarsa) dan peristiwa (seperti kedaluwarsa materi kunci yang diimpor atau penghapusan atau rotasi kunci kunci KMS).

Alat-alat pemantauan

AWSmenyediakan berbagai alat yang dapat Anda gunakan untuk memantau kunci KMS Anda. Anda dapat mengonfigurasi beberapa alat ini untuk melakukan pemantauan untuk Anda, sementara beberapa alat memerlukan intervensi manual. Kami menyarankan agar Anda mengautomasi tugas pemantauan sebanyak mungkin.

Alat pemantauan otomatis

Anda dapat menggunakan alat pemantauan otomatis berikut untuk melihat kunci KMS Anda dan melaporkan ketika sesuatu telah berubah.

- AWS CloudTrailPemantauan Log - Bagikan file log antar akun, pantau file CloudTrail log secara real time dengan mengirimkannya ke CloudWatch Log, menulis aplikasi pemrosesan log dengan Pustaka CloudTrail Pemrosesan, dan validasi bahwa file log Anda tidak berubah setelah pengiriman oleh CloudTrail. Untuk informasi selengkapnya, lihat Bekerja dengan File CloudTrail Log di Panduan AWS CloudTrail Pengguna.
- CloudWatch Alarm Amazon — Tonton satu metrik selama periode waktu yang Anda tentukan, dan lakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Tindakannya adalah pemberitahuan yang dikirim ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Amazon EC2 Auto Scaling. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu; negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu. Untuk informasi selengkapnya, lihat Pemantauan CloudWatch dengan Amazon.
- Amazon EventBridge — Cocokkan acara dan arahkan ke satu atau beberapa fungsi atau aliran target untuk menangkap informasi status dan, jika perlu, membuat perubahan atau mengambil tindakan korektif. Untuk informasi selengkapnya, lihat Pemantauan EventBridge dengan Amazon dan Panduan EventBridge Pengguna Amazon.
- Amazon CloudWatch Logs — Pantau, simpan, dan akses file log Anda dari AWS CloudTrail atau sumber lain. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.

Alat pemantauan manual

Bagian penting lainnya dari pemantauan kunci KMS melibatkan pemantauan secara manual item yang tidak CloudWatch tercakup oleh alarm dan peristiwa. AWSDasbor AWS KMS CloudWatchAWS Trusted Advisor,,, dan lainnya memberikan at-a-glance pandangan tentang keadaan AWS lingkungan Anda.

Anda dapat menyesuaikan halaman kunci yang dikelola Pelanggan di AWS KMSkonsol untuk menampilkan informasi berikut tentang setiap kunci KMS: Kunci yang dikelola AWS

- ID Kunci
- Status
- Tanggal pembuatan

- Tanggal kedaluwarsa (untuk kunci KMS dengan bahan kunci impor)
- Asal
- ID toko kunci kustom (untuk kunci KMS di toko kunci khusus)

Dasbor CloudWatch konsol menunjukkan hal berikut:

- Alarm dan status saat ini
- Grafik alarm dan sumber daya
- Status kesehatan layanan

Selain itu, Anda dapat menggunakan CloudWatch untuk melakukan hal berikut:

- Membuat dasbor yang disesuaikan untuk memantau layanan yang penting bagi Anda
- Grafik data metrik untuk memecahkan masalah dan menemukan tren
- Cari dan telusuri semua metrik sumber daya AWS Anda
- Membuat dan mengedit alarm untuk menerima notifikasi tentang masalah

AWS Trusted Advisor dapat membantu memantau sumber daya AWS Anda untuk meningkatkan performa, keandalan, keamanan, dan efektivitas biaya. Empat pemeriksaan Trusted Advisor tersedia bagi semua pengguna; lebih dari 50 pemeriksaan tersedia bagi pengguna dengan perencanaan dukungan Bisnis atau Korporasi. Untuk informasi selengkapnya, lihat AWS Trusted Advisor.

Logging panggilan AWS KMS API dengan AWS CloudTrail

AWS KMS terintegrasi dengan AWS CloudTrail, layanan yang merekam semua panggilan AWS KMS oleh pengguna, peran, dan AWS layanan lainnya. CloudTrail menangkap semua panggilan API ke AWS KMS sebagai event, termasuk panggilan dari AWS KMS konsol, AWS KMS API, AWS CloudFormation template, AWS Command Line Interface (AWS CLI), dan AWS Tools for PowerShell.

CloudTrail mencatat semua AWS KMS operasi, termasuk operasi hanya-baca, seperti ListAliasesdan GetKeyRotationStatus, operasi yang mengelola kunci KMS, seperti dan, CreateKeydan operasi kriptografi PutKeyPolicy, seperti dan Dekripsi. GenerateDataKey Ini juga mencatat operasi internal yang AWS KMS memanggil Anda, seperti DeleteExpiredKeyMaterial, DeleteKey, SynchronizeMultiRegionKey, dan RotateKey.

CloudTrail mencatat operasi yang berhasil dan percobaan panggilan yang gagal, seperti ketika pemanggil ditolak akses ke sumber daya. Operasi lintas akun pada kunci KMS dicatat di akun penelepon dan akun pemilik kunci KMS. Namun, AWS KMS permintaan lintas akun yang ditolak karena akses ditolak hanya dicatat di akun pemanggil.

Untuk alasan keamanan, beberapa bidang dihilangkan dari entri AWS KMS log, seperti `Plaintext` parameter permintaan Enkripsi, dan respons terhadap GetKeyPolicyatau operasi kriptografi apa pun. Untuk mempermudah pencarian entri CloudTrail log untuk kunci KMS tertentu, AWS KMS tambahkan ARN kunci dari kunci KMS yang terpengaruh ke `responseElements` bidang di entri log untuk beberapa operasi manajemen AWS KMS kunci, bahkan ketika operasi API tidak mengembalikan ARN kunci.

Meskipun secara default, semua AWS KMS tindakan dicatat sebagai CloudTrail peristiwa, Anda dapat mengecualikan AWS KMS tindakan dari CloudTrail jejak. Lihat perinciannya di Tidak termasuk AWS KMS acara dari jejak.

Pelajari lebih lanjut:

- Untuk contoh CloudTrail log AWS KMS operasi untuk kantong AWS Nitro, lihat. Memantau permintaan untuk kantong Nitro

Topik

- Logging peristiwa di CloudTrail
- Mencari acara di CloudTrail
- Tidak termasuk AWS KMS acara dari jejak
- Contoh entri AWS KMS log

Logging peristiwa di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di AWS KMS, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk AWS KMS, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS.

Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi lainnya Layanan AWS untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Lihat informasi yang lebih lengkap di:

- Ikhtisar untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa Wilayah dan Menerima file CloudTrail log dari beberapa akun

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna. Untuk mempelajari cara lain untuk memantau penggunaan tombol KMS Anda, lihatMemantau AWS KMS keys.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Jika permintaan dibuat dengan kredensi root atau kredensil pengguna IAM.
- Jika permintaan dibuat dengan kredensil keamanan sementara untuk peran atau pengguna federasi.
- Jika permintaan itu dibuat oleh orang lain Layanan AWS.

Untuk informasi selengkapnya, lihat Elemen CloudTrail UserIdentity.

Mencari acara di CloudTrail

Untuk mencari entri CloudTrail log, gunakan CloudTrail konsol atau CloudTrail LookupEventsoperasi. CloudTrail mendukung banyak nilai atribut untuk memfilter pencarian Anda, termasuk nama acara, nama pengguna, dan sumber acara.

Untuk membantu Anda mencari entri AWS KMS log di CloudTrail, AWS KMS isi kolom entri CloudTrail log berikut.

> **Note**
>
> Mulai Desember 2022, AWS KMS mengisi atribut tipe Sumber Daya dan nama Sumber Daya di semua operasi manajemen yang mengubah kunci KMS tertentu. Nilai atribut

ini mungkin null dalam CloudTrail entri lama untuk operasi berikut: CreateAlias,, CreateGrant, DeleteAlias, DeleteImportedKeyMaterial, ImportKeyMaterial, ReplicateKey, RetireGrantRevokeGrantUpdateAlias, dan. UpdatePrimaryRegion

Atribut	Nilai	Entri log
Sumber acara (`EventSource`)	`kms.amazonaws.com`	Semua operasi.
Jenis sumber daya (`ResourceType`)	`AWS::KMS::Key`	Operasi manajemen yang mengubah kunci KMS tertentu, seperti `CreateKey` dan`EnableKey`, tetapi tidak`ListKeys`.
Nama sumber daya (`ResourceName`)	ARN kunci (atau ID kunci dan kunci ARN)	Operasi manajemen yang mengubah kunci KMS tertentu, seperti `CreateKey` dan`EnableKey`, tetapi tidak`ListKeys`.

Untuk membantu Anda menemukan entri log untuk operasi manajemen pada kunci KMS tertentu, AWS KMS mencatat ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` elemen entri log, bahkan ketika operasi AWS KMS API tidak mengembalikan kunci ARN.

Misalnya, panggilan yang berhasil ke DisableKeyoperasi tidak mengembalikan nilai apa pun dalam respons, tetapi alih-alih nilai nol, `responseElements.keyId` nilai dalam entri DisableKey log menyertakan ARN kunci dari kunci KMS yang dinonaktifkan.

Fitur ini ditambahkan pada Desember 2022 dan memengaruhi entri CloudTrail log berikut: CreateAlias,,,, CreateGrant, DeleteAlias, DeleteKey, DisableKey, EnableKey, EnableKeyRotation, ImportKeyMaterial, RotateKey, SynchronizeMultiRegionKey, TagResource, UntagResourceUpdateAlias, dan UpdatePrimaryRegion.

Tidak termasuk AWS KMS acara dari jejak

Untuk memberikan catatan penggunaan dan pengelolaan AWS KMS sumber daya mereka, sebagian besar AWS KMS pengguna mengandalkan peristiwa dalam CloudTrail jejak. Jejak dapat menjadi sumber data yang berharga untuk mengaudit peristiwa penting, seperti membuat, menonaktifkan, dan menghapus, mengubah kebijakan utama AWS KMS keys, dan penggunaan kunci KMS Anda oleh AWS layanan atas nama Anda. Dalam beberapa kasus, metadata dalam entri CloudTrail log, seperti konteks enkripsi dalam operasi enkripsi, dapat membantu Anda menghindari atau menyelesaikan kesalahan.

Namun, karena AWS KMS dapat menghasilkan sejumlah besar peristiwa, AWS CloudTrail memungkinkan Anda mengecualikan AWS KMS acara dari jejak. Pengaturan per-jejak ini mengecualikan semua AWS KMS peristiwa; Anda tidak dapat mengecualikan acara tertentu. AWS KMS

> **Warning**
>
> Mengecualikan AWS KMS peristiwa dari CloudTrail Log dapat mengaburkan tindakan yang menggunakan kunci KMS Anda. Berhati-hatilah saat memberikan `cloudtrail:PutEventSelectors` kepada pengguna utama yang diperlukan untuk melakukan operasi ini.

Untuk mengecualikan AWS KMS acara dari jejak:

- Di CloudTrail konsol, gunakan setelan peristiwa Layanan Manajemen Kunci Log saat Anda membuat jejak atau memperbarui jejak. Untuk petunjuknya, lihat Logging Management Events dengan AWS Management Console di Panduan AWS CloudTrail Pengguna.
- Di CloudTrail API, gunakan PutEventSelectorsoperasi. Tambahkan atribut `ExcludeManagementEventSources` pada pemilih peristiwa Anda dengan nilai `kms.amazonaws.com`. Sebagai contoh, lihat Contoh: Jejak yang tidak mencatat AWS Key Management Service peristiwa di Panduan AWS CloudTrail Pengguna.

Anda dapat menonaktifkan pengecualian ini kapan saja dengan mengubah pengaturan konsol atau pemilih peristiwa untuk jejak. Jejak kemudian akan mulai merekam AWS KMS acara. Namun, itu tidak dapat memulihkan AWS KMS peristiwa yang terjadi saat pengecualian efektif.

Saat Anda mengecualikan AWS KMS peristiwa dengan menggunakan konsol atau API, operasi CloudTrail `PutEventSelectors` API yang dihasilkan juga dicatat di CloudTrail Log Anda. Jika AWS KMS peristiwa tidak muncul di CloudTrail Log Anda, cari `PutEventSelectors` acara dengan `ExcludeManagementEventSources` atribut yang disetel ke`kms.amazonaws.com`.

Contoh entri AWS KMS log

AWS KMS menulis entri ke CloudTrail log Anda saat Anda memanggil AWS KMS operasi dan ketika AWS layanan memanggil operasi atas nama Anda. AWS KMS juga menulis entri ketika memanggil operasi untuk Anda. Misalnya, ia menulis entri ketika menghapus kunci KMS yang Anda jadwalkan untuk dihapus.

Topik berikut menampilkan contoh entri CloudTrail log untuk AWS KMS operasi.

Untuk contoh entri CloudTrail log permintaan AWS KMS dari AWS Nitro Enclave, lihat. Memantau permintaan untuk kantong Nitro

Topik

- CancelKeyDeletion
- ConnectCustomKeyStore
- CreateAlias
- CreateCustomKeyStore
- CreateGrant
- CreateKey
- Dekripsi
- DeleteAlias
- DeleteCustomKeyStore
- DeleteExpiredKeyMaterial
- DeleteImportedKeyMaterial
- DeleteKey
- DescribeCustomKeyStores
- DescribeKey
- DisableKey
- DisableKeyRotation
- DisconnectCustomKeyStore

- EnableKey
- EnableKeyRotation
- Enkripsi
- 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
- Verifikasi
- Contoh Amazon EC2
- Contoh dua Amazon EC2

CancelKeyDeletion

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil CancelKeyDeletionoperasi. Untuk informasi tentang menghapusAWS KMS keys, lihatMenghapus 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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `ConnectCustomKeyStore`operasi. Untuk informasi tentang menghubungkan toko kunci kustom, lihatMenghubungkan dan memutuskan sambungan toko AWS CloudHSM kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk CreateAliasoperasi. `resources`Elemen ini mencakup bidang untuk alias dan sumber daya kunci KMS. Untuk informasi tentang cara membuat alias di AWS KMS, lihat Membuat alias.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `CreateCustomKeyStore`operasi di toko AWS CloudHSM kunci. Untuk informasi tentang membuat toko kunci kustom, lihatMembuat toko AWS CloudHSM kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk CreateGrantoperasi. Untuk informasi tentang cara membuat izin di AWS KMS, lihat Hibah di AWS KMS.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan ARN kunci.

```
{
  "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

Contoh-contoh ini menunjukkan entri AWS CloudTrail log untuk CreateKeyoperasi.

Entri `CreateKey` log dapat dihasilkan dari `CreateKey` permintaan atau `CreateKey` operasi untuk ReplicateKeypermintaan.

Contoh berikut menunjukkan entri CloudTrail log untuk CreateKeyoperasi yang menciptakan kunci KMS enkripsi simetris. Untuk informasi tentang membuat kunci KMS, lihatMembuat kunci.

```
{
    "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"
}
```

Contoh berikut menunjukkan CloudTrail log `CreateKey` operasi yang menciptakan kunci KMS enkripsi simetris di toko AWS CloudHSMkunci.

```
{
    "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"
}
```


Contoh berikut menunjukkan CloudTrail log `CreateKey` operasi yang membuat kunci KMS enkripsi simetris di toko kunci eksternal.


```
{
```

```
    "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"
}
```


Dekripsi

Contoh-contoh ini menunjukkan entri AWS CloudTrail log untuk operasi Dekripsi.

Entri CloudTrail log untuk `Decrypt` operasi selalu menyertakan `encryptionAlgorithm` dalam `requestParameters` bahkan jika algoritma enkripsi tidak ditentukan dalam permintaan. Ciphertext dalam permintaan dan plaintext dalam respons dihilangkan.

Topik

- Dekripsi dengan kunci enkripsi simetris standar
- Dekripsi kegagalan dengan kunci enkripsi simetris standar
- Dekripsi dengan kunci KMS di toko kunci AWS CloudHSM
- Dekripsi dengan kunci KMS di toko kunci eksternal
- Dekripsi kegagalan dengan kunci KMS di toko kunci eksternal

Dekripsi dengan kunci enkripsi simetris standar

Berikut ini adalah contoh entri CloudTrail log untuk `Decrypt` operasi dengan kunci enkripsi simetris standar.


```
{
    "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"
```

```
}
```

Dekripsi kegagalan dengan kunci enkripsi simetris standar

Contoh entri CloudTrail log berikut mencatat `Decrypt` operasi yang gagal dengan kunci KMS enkripsi simetris standar. Exception (`errorCode`) dan error message (`errorMessage`) disertakan membantu Anda mengatasi kesalahan.

Dalam hal ini, kunci KMS enkripsi simetris yang ditentukan dalam `Decrypt` permintaan bukanlah kunci KMS enkripsi simetris yang digunakan untuk mengenkripsi data.

```
{
    "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"
}
```

Dekripsi dengan kunci KMS di toko kunci AWS CloudHSM

Contoh entri CloudTrail log berikut mencatat `Decrypt` operasi dengan kunci KMS di toko AWS CloudHSM kunci. Semua entri log untuk operasi kriptografi dengan kunci KMS di toko kunci khusus menyertakan `additionalEventData` bidang dengan file. `customKeyStoreId` `additionalEventData`Tidak ditentukan dalam permintaan.

```
{
    "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"
}
```

Dekripsi dengan kunci KMS di toko kunci eksternal

Contoh entri CloudTrail log berikut mencatat `Decrypt` operasi dengan kunci KMS di toko kunci eksternal. Selain itu`customKeyStoreId`, `additionalEventData` bidang termasuk ID kunci eksternal (`XksKeyId`). `additionalEventData`Tidak ditentukan dalam permintaan.

```
{
    "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"
}
```

Dekripsi kegagalan dengan kunci KMS di toko kunci eksternal

Contoh entri CloudTrail log berikut mencatat permintaan gagal untuk `Decrypt` operasi dengan kunci KMS di penyimpanan kunci eksternal. CloudWatch mencatat permintaan yang gagal, selain permintaan yang berhasil. Saat merekam kegagalan, entri CloudTrail log menyertakan pengecualian (ErrorCode) dan pesan kesalahan yang menyertainya (ErrorMessage).

Jika permintaan gagal mencapai proxy penyimpanan kunci eksternal Anda, seperti dalam contoh ini, Anda dapat menggunakan `requestId` nilai untuk mengaitkan permintaan yang gagal dengan permintaan yang sesuai, log proxy penyimpanan kunci eksternal Anda, jika proxy Anda menyediakannya.

Untuk bantuan terkait `Decrypt` permintaan di toko kunci eksternal, lihatKesalahan dekripsi.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk DeleteAliasoperasi. Untuk informasi tentang cara menghapus alias, lihat Menghapus alias.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan ARN kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `DeleteCustomKeyStore`operasi. Untuk informasi tentang membuat toko kunci kustom, lihatMenghapus toko AWS CloudHSM kunci.

```
{
    "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

Saat Anda mengimpor materi kunci ke dalam AWS KMS key (kunci KMS), Anda dapat mengatur tanggal dan waktu kedaluwarsa untuk materi kunci tersebut. AWS KMSmencatat entri di CloudTrail log Anda saat Anda mengimpor materi kunci (dengan pengaturan kedaluwarsa) dan saat AWS KMS menghapus materi kunci yang kedaluwarsa. Untuk informasi tentang membuat kunci KMS dengan materi kunci impor, lihatMengimpor bahan kunci untuk AWS KMS kunci.

Contoh berikut menunjukkan entri log AWS CloudTrail yang dihasilkan ketika AWS KMS menghapus materi kunci kadaluwarsa.


```
{
    "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

Jika Anda mengimpor materi kunci ke kunci KMS, Anda dapat menghapus materi kunci yang diimpor kapan saja dengan menggunakan DeleteImportedKeyMaterialoperasi. Saat Anda menghapus materi kunci yang diimpor dari kunci KMS, status kunci kunci KMS berubah menjadi `PendingImport` dan kunci KMS tidak dapat digunakan dalam operasi kriptografi apa pun. Untuk detailnya, lihat Menghapus material kunci yang diimpor.

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan untuk `DeleteImportedKeyMaterial` operasi.

```
{
    "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

Contoh-contoh ini menunjukkan entri AWS CloudTrail log yang dihasilkan ketika kunci KMS dihapus. Untuk menghapus kunci KMS, Anda menggunakan ScheduleKeyDeletionoperasi. Setelah masa tunggu yang ditentukan berakhir, AWS KMS hapus kunci KMS dan catat entri seperti yang berikut di CloudTrail log Anda untuk merekam peristiwa itu.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

Untuk contoh entri CloudTrail log untuk `ScheduleKeyDeletion` operasi, lihatScheduleKeyDeletion. Untuk informasi tentang menghapus kunci KMS, lihat. Menghapus AWS KMS keys

Contoh entri CloudTrail log berikut mencatat `DeleteKey` operasi kunci KMS dengan materi kunci diAWS 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"
}
```

Entri CloudTrail log berikut mencatat `DeleteKey` operasi kunci KMS di [toko kunci AWS CloudHSM kustom](#).

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `DescribeCustomKeyStores`operasi. Untuk informasi tentang melihat toko kunci khusus, lihatMelihat toko AWS CloudHSM kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk DescribeKeyoperasi. AWS KMSmerekam entri seperti yang berikut ketika Anda memanggil `DescribeKey` operasi atau melihat kunci KMS di AWS KMS konsol. Panggilan ini adalah hasil dari melihat kunci di konsol AWS KMS manajemen.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk DisableKeyoperasi. Untuk informasi tentang mengaktifkan dan menonaktifkan, lihatAWS KMS keys. AWS KMS Mengaktifkan dan menonaktifkan kunci

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil DisableKeyRotationoperasi. Untuk informasi tentang rotasi tombol otomatis, lihatBerputar 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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `DisconnectCustomKeyStore`operasi. Untuk informasi tentang memutuskan sambungan penyimpanan kunci kustom, lihatMenghubungkan dan memutuskan sambungan toko AWS CloudHSM kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk EnableKeyoperasi. Untuk informasi tentang mengaktifkan dan menonaktifkan, lihatAWS KMS keys.. AWS KMS Mengaktifkan dan menonaktifkan kunci

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log panggilan ke EnableKeyRotationoperasi. Untuk contoh entri CloudTrail log yang ditulis saat kunci diputar, lihatRotateKey. Untuk informasi tentang rotasi AWS KMS keys, lihatBerputar AWS KMS keys.

> **Note**
>
> rotation-periodIni adalah parameter permintaan opsional. Jika Anda tidak menentukan periode rotasi saat Anda mengaktifkan rotasi kunci otomatis, nilai defaultnya adalah 365 hari.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.


```
{
    "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"
}
```

Enkripsi

Contoh berikut menunjukkan entri log AWS CloudTrail untuk operasi Enkripsi.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyoperasi.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyPairoperasi. Contoh ini mencatat operasi yang menghasilkan key pair RSA yang dienkripsi di bawah enkripsi simetris. AWS KMS key

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyPairWithoutPlaintextoperasi. Contoh ini mencatat operasi yang menghasilkan key pair RSA yang dienkripsi di bawah enkripsi simetris. AWS KMS key

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateDataKeyWithoutPlaintextoperasi.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateMacoperasi.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GenerateRandomoperasi. Karena operasi ini tidak menggunakanAWS KMS key, `resources` bidang kosong.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GetKeyPolicyoperasi. Untuk informasi tentang melihat kebijakan kunci untuk kunci KMS, lihatMelihat kebijakan kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk GetKeyRotationStatusoperasi. Untuk informasi tentang rotasi otomatis dan sesuai permintaan bahan kunci untuk kunci KMS, lihat. Berputar 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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan saat Anda menggunakan GetParametersForImportoperasi. Operasi ini mengembalikan kunci publik dan token impor yang Anda gunakan saat mengimpor materi kunci ke kunci KMS. CloudTrail Entri yang sama direkam saat Anda menggunakan `GetParametersForImport` operasi atau menggunakan AWS KMS konsol untuk mengunduh kunci publik dan token impor.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan saat Anda menggunakan ImportKeyMaterialoperasi. CloudTrail Entri yang sama direkam saat Anda menggunakan `ImportKeyMaterial` operasi atau menggunakan AWS KMS konsol untuk mengimpor materi kunci ke dalam fileAWS KMS key.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ListAliasesoperasi. Karena operasi ini tidak menggunakan alias tertentu atauAWS KMS key, `resources` bidang kosong. Untuk informasi tentang cara melihat alias di AWS KMS, lihat Melihat alias.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ListGrantoperasi. Untuk informasi tentang izin di AWS KMS, lihat Hibah di 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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ListKeyRotationsoperasi. Untuk informasi tentang rotasi otomatis dan sesuai permintaan bahan kunci untuk kunci KMS, lihat. Berputar 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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil PutKeyPolicyoperasi. Untuk informasi tentang memperbarui kebijakan utama, lihatMengubah kebijakan kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk ReEncryptoperasi. `resources`Bidang dalam entri log ini menentukan duaAWS KMS keys, kunci KMS sumber dan kunci KMS tujuan, dalam urutan itu.


```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil ReplicateKeyoperasi. `ReplicateKey`Permintaan menghasilkan `ReplicateKey` operasi dan CreateKeyoperasi.

Untuk informasi tentang cara mereplikasi kunci multi-Wilayah, lihat Membuat kunci replika multi-Wilayah.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil RetireGrantoperasi. Untuk informasi tentang pensiun hibah, lihat. Menghentikan dan mencabut pemberian izin


```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil RevokeGrantoperasi. Untuk informasi tentang mencabut hibah, lihat. Menghentikan dan mencabut pemberian izin


```
{
    "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

Contoh-contoh ini menunjukkan entri AWS CloudTrail log untuk operasi yang berputar AWS KMS keys. Untuk informasi tentang memutar tombol KMS, lihat. Berputar AWS KMS keys

Contoh berikut menunjukkan entri CloudTrail log untuk operasi yang memutar kunci KMS enkripsi simetris di mana rotasi kunci otomatis diaktifkan. Untuk informasi tentang mengaktifkan rotasi otomatis, lihatCara mengaktifkan dan menonaktifkan rotasi kunci otomatis.

Untuk contoh entri CloudTrail log yang mencatat `EnableKeyRotation` operasi, lihatEnableKeyRotation.


```
{
    "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"
```

```
}
```

Contoh berikut menunjukkan entri CloudTrail log untuk RotateKeyOnDemandoperasi. Untuk informasi tentang memutar kunci KMS enkripsi simetris sesuai permintaan, lihat. Cara melakukan rotasi kunci sesuai permintaan

Untuk contoh entri CloudTrail log yang mencatat `RotateKeyOnDemand` operasi, lihatRotateKeyOnDemand.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk RotateKeyOnDemandoperasi. Untuk contoh entri CloudTrail log yang ditulis saat kunci diputar, lihatRotateKey. Untuk informasi lebih lanjut tentang rotasi sesuai permintaan bahan kunci untuk kunci KMS, lihat. Cara melakukan rotasi kunci sesuai permintaan


```
{
    "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

Contoh-contoh ini menunjukkan entri AWS CloudTrail log untuk ScheduleKeyDeletionoperasi.

Untuk contoh entri CloudTrail log yang ditulis saat kunci dihapus, lihatDeleteKey. Untuk informasi tentang menghapusAWS KMS keys, lihatMenghapus AWS KMS keys.

Contoh berikut mencatat `ScheduleKeyDeletion` permintaan untuk kunci KMS Single-region.

```
{
    "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"
}
```

Contoh berikut mencatat `ScheduleKeyDeletion` permintaan untuk kunci KMS Multi-wilayah dengan kunci replika.

Karena AWS KMS tidak akan menghapus kunci Multi-wilayah sampai semua kunci replika dihapus, di `responseElements` bidang, `keyState` is `PendingReplicaDeletion` dan `deletionDate` bidang dihilangkan.

```
{
    "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"
}
```

Contoh berikut mencatat `ScheduleKeyDeletion` permintaan untuk kunci KMS di [toko kunci AWS CloudHSM kustom](#).

```
{
    "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

Contoh-contoh ini menunjukkan entri AWS CloudTrail log untuk operasi Tanda tangan.

Contoh berikut menunjukkan entri CloudTrail log untuk operasi Tanda yang menggunakan kunci KMS RSA asimetris untuk menghasilkan tanda tangan digital untuk sebuah file.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:36:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Sign",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "messageType": "RAW",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
    },
    "responseElements": null,
    "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
    "eventID": "107b3cac-b125-4556-9702-12a2b9afcff7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


SynchronizeMultiRegionKey

Contoh berikut menunjukkan entri log AWS CloudTrail dihasilkan ketika AWS KMS menyinkronkan kunci multi-Wilayah. Sinkronisasi melibatkan panggilan lintas Wilayah untuk menyalin properti bersama dari kunci utama multi-Wilayah untuk kunci replikanya. AWS KMS akan menyinkronkan kunci multi-Wilayah secara berkala untuk memastikan bahwa semua kunci multi-Wilayah terkait memiliki materi kunci yang sama.

`resources`Elemen entri CloudTrail log mencakup ARN kunci dari kunci primer Multi-wilayah, termasuk nya. Wilayah AWS Kunci replika multi-Wilayah terkait dan Wilayah mereka tidak tercantum dalam entri log ini.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log panggilan ke TagResourceoperasi untuk menambahkan tag dengan kunci tag `Department` dan nilai tag dari`IT`.

Untuk contoh entri `UntagResource` CloudTrail log yang ditulis saat kunci diputar, lihatUntagResource. Untuk informasi tentang penandaanAWS KMS keys, lihatTombol penandaan.


```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log panggilan ke UntagResourceoperasi untuk menghapus tag dengan kunci tag dari`Dept`.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

Untuk contoh entri `TagResource` CloudTrail log, lihatTagResource. Untuk informasi tentang penandaanAWS KMS keys, lihatTombol penandaan.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk UpdateAliasoperasi. `resources`Elemen ini mencakup bidang untuk alias dan sumber daya kunci KMS. Untuk informasi tentang cara membuat alias di AWS KMS, lihat Membuat alias.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan kunci ARN.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil `UpdateCustomKeyStore`operasi untuk memperbarui ID cluster untuk penyimpanan kunci kustom. Untuk informasi tentang mengedit toko kunci kustom, lihatMengedit pengaturan toko AWS CloudHSM kunci.

```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil UpdateKeyDescriptionoperasi.


```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log yang dihasilkan dengan memanggil UpdatePrimaryRegionoperasi pada kunci Multi-region.

`UpdatePrimaryRegion`Operasi menulis dua entri CloudTrail log: satu di Wilayah dengan kunci primer Multi-wilayah yang dikonversi menjadi kunci replika, dan satu di Wilayah dengan kunci replika Multi-wilayah yang dikonversi ke kunci primer.

CloudTrail entri log untuk operasi ini yang direkam pada atau setelah Desember 2022 menyertakan ARN kunci dari kunci KMS yang terpengaruh dalam `responseElements.keyId` nilainya, meskipun operasi ini tidak mengembalikan ARN kunci.

Contoh berikut menunjukkan entri CloudTrail log untuk `UpdatePrimaryRegion` di Wilayah di mana kunci Multi-region berubah dari kunci primer menjadi kunci replika (us-barat-2). Kolom `primaryRegion` menunjukkan Wilayah yang sekarang menghosting kunci utama (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"
}
```

Contoh berikut mewakili entri CloudTrail log untuk `UpdatePrimaryRegion` di Wilayah di mana kunci Multi-wilayah berubah dari kunci replika ke kunci utama (ap-northeast-1). Entri log ini tidak mengidentifikasi Wilayah utama sebelumnya.


```
{
    "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

Contoh berikut menunjukkan entri AWS CloudTrail log untuk VerifyMacoperasi.

```
{
    "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"
}
```


Verifikasi

Contoh ini menunjukkan entri AWS CloudTrail log untuk operasi Verifikasi.

Contoh berikut menunjukkan entri CloudTrail log untuk operasi Verifikasi yang menggunakan kunci KMS RSA asimetris untuk memverifikasi tanda tangan digital.

```
{
    "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"
}
```


Contoh Amazon EC2

Contoh berikut mencatat prinsipal IAM yang membuat volume terenkripsi menggunakan kunci volume default di konsol manajemen Amazon EC2.

Contoh berikut menunjukkan entri CloudTrail log di mana pengguna Alice membuat volume terenkripsi dengan kunci volume default di konsol manajemen Amazon EC2. Catatan berkas log EC2 mencakup bidang `volumeId` dengan nilai `"vol-13439757"`. Catatan AWS KMS berisi bidang `encryptionContext` dengan nilai `"aws:ebs:id": "vol-13439757"`. Demikian pula, `principalId` dan `accountId` antara dua catatan cocok. Catatan mencerminkan fakta bahwa membuat volume terenkripsi menghasilkan kunci data yang digunakan untuk mengenkripsi konten volume.


```
{
  "Records": [
     {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
        "size": "10",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1415220618876,
        "volumeType": "gp2",
        "iops": 30,
        "encrypted": true
      },
```

```
      "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
      "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
      "eventType": "AwsApiCall",
      "recipientAccountId": "123456789012"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:19Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "&AWS; Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-13439757"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-123456789012-758241111-1415220618",
      "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
```

```
}
```

Contoh dua Amazon EC2

Dalam contoh berikut, prinsipal IAM yang menjalankan instans Amazon EC2 membuat dan memasang volume data yang dienkripsi di bawah kunci KMS. Tindakan ini menghasilkan beberapa catatan CloudTrail log.

Ketika volume dibuat, Amazon EC2, yang bertindak atas nama pelanggan, mendapatkan kunci data terenkripsi dari AWS KMS (`GenerateDataKeyWithoutPlaintext`). Kemudian, volume membuat izin (`CreateGrant`) yang memungkinkannya untuk mendekripsi kunci data. Ketika volume terpasang, Amazon EC2 memanggil AWS KMS untuk mendekripsi kunci data (`Decrypt`).

`instanceId` dari instans Amazon EC2, `"i-81e2f56c"`, akan muncul dalam peristiwa `RunInstances`. ID instans yang sama memenuhi syarat `granteePrincipal` dari izin yang dibuat (`"111122223333:aws:ec2-infrastructure:i-81e2f56c"`) dan asumsi peran yang merupakan perwakilan dalam panggilan `Decrypt` (`"arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c"`).

Kunci ARN dari kunci KMS yang melindungi volume data,`arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, muncul di ketiga AWS KMS panggilan (`CreateGrant`,`GenerateDataKeyWithoutPlaintext`, dan). `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"
    }
  ]
}
```

Pemantauan CloudWatch dengan Amazon

Anda dapat memantau AWS KMS keys penggunaan Amazon CloudWatch, AWS layanan yang mengumpulkan dan memproses data mentah dari AWS KMS metrik yang dapat dibaca, mendekati waktu nyata. Data ini direkam untuk jangka waktu dua minggu sehingga Anda dapat mengakses informasi historis dan mendapatkan pemahaman yang lebih baik tentang penggunaan kunci KMS Anda dan perubahannya dari waktu ke waktu.

Anda dapat menggunakan Amazon CloudWatch untuk mengingatkan Anda tentang peristiwa penting, seperti yang berikut.

- Materi kunci yang diimpor dalam kunci KMS mendekati tanggal kedaluwarsanya.
- Kunci KMS yang tertunda penghapusan masih digunakan.
- Materi kunci dalam kunci KMS diputar secara otomatis.
- Kunci KMS telah dihapus.

Anda juga dapat membuat CloudWatch alarm Amazon yang memberi tahu Anda ketika tingkat permintaan Anda mencapai persentase tertentu dari nilai kuota. Untuk detailnya, lihat Mengelola tarif

permintaan AWS KMS API Anda menggunakan Service Quotas dan Amazon CloudWatch di Blog AWS Keamanan.

Topik

- AWS KMS metrik dan dimensi
- Melihat AWS KMS metrik
- Membuat CloudWatch alarm untuk memantau tombol KMS

AWS KMS metrik dan dimensi

AWS KMS menetapkan CloudWatch metrik Amazon untuk memudahkan Anda memantau data penting dan membuat alarm. Anda dapat melihat AWS KMS metrik menggunakan AWS Management Console dan Amazon CloudWatch API.

Bagian ini mencantumkan setiap AWS KMS metrik dan dimensi untuk setiap metrik, dan memberikan beberapa panduan dasar untuk membuat CloudWatch alarm berdasarkan metrik dan dimensi ini.

> **Note**
>
> Nama grup dimensi:
> Untuk melihat metrik di CloudWatch konsol Amazon, di bagian Metrik, pilih nama grup dimensi. Kemudian Anda dapat memfilter dengan nama Metrik. Topik ini mencakup nama metrik dan nama grup dimensi untuk setiap AWS KMS metrik.

Topik

- SecondsUntilKeyMaterialExpiration
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors
- XksExternalKeyManagerStates
- XksProxyLatency

SecondsUntilKeyMaterialExpiration

Jumlah detik yang tersisa sampai bahan kunci yang diimpor dalam kunci KMS kedaluwarsa. Metrik ini hanya berlaku untuk kunci KMS dengan bahan kunci impor (asal bahan utamaEXTERNAL) dan tanggal kedaluwarsa.

Gunakan metrik ini untuk melacak waktu yang tersisa hingga materi kunci impor Anda kedaluwarsa. Ketika waktu itu jatuh di bawah ambang batas yang Anda tentukan, Anda mungkin ingin mengimpor ulang materi kunci dengan tanggal kedaluwarsa baru. `SecondsUntilKeyMaterialExpiration`Metrik khusus untuk kunci KMS. Anda tidak dapat menggunakan metrik ini untuk memantau beberapa kunci KMS atau kunci KMS yang mungkin Anda buat di masa mendatang. Untuk bantuan dalam membuat CloudWatch alarm untuk memantau metrik ini, lihatMembuat CloudWatch alarm untuk kedaluwarsa materi kunci yang diimpor.

Statistik yang paling berguna untuk metrik ini adalah`Minimum`, yang memberi tahu Anda jumlah waktu terkecil yang tersisa untuk semua titik data dalam periode statistik yang ditentukan. Satu-satunya unit yang valid untuk metrik ini adalah `Seconds`.

Nama grup dimensi: Metrik Per-Kunci

Dimensi untuk **`SecondsUntilKeyMaterialExpiration`**

Dimensi	Deskripsi; terkait dengan AWS
KeyId	Nilai untuk setiap kunci KMS.

ExternalKeyStoreThrottle

Jumlah permintaan untuk operasi kriptografi pada kunci KMS di setiap penyimpanan kunci eksternal yang AWS KMS melambat (merespons dengan a). `ThrottlingException` Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

`ExternalKeyStoreThrottle`Metrik hanya berlaku untuk kunci KMS di penyimpanan kunci eksternal dan hanya untuk permintaan operasi kriptografi dan operasi. DescribeKey AWS KMS membatasi permintaan ini ketika tingkat permintaan melebihi kuota permintaan toko kunci kustom untuk penyimpanan kunci eksternal Anda. Metrik ini tidak termasuk pembatasan oleh proxy penyimpanan kunci eksternal atau pengelola kunci eksternal Anda.

Gunakan metrik ini untuk meninjau dan menyesuaikan nilai kuota permintaan toko kunci kustom Anda. Jika metrik ini menunjukkan bahwa AWS KMS sering membatasi permintaan Anda untuk

kunci KMS ini, Anda dapat mempertimbangkan untuk meminta peningkatan nilai kuota permintaan penyimpanan kunci kustom Anda. Untuk bantuan, lihat Meminta peningkatan kuota dalam Panduan Pengguna Service Quotas.

Jika Anda mendapatkan `KMSInvalidStateException` kesalahan yang sangat sering dengan pesan yang menjelaskan bahwa permintaan ditolak "karena tingkat permintaan yang sangat tinggi" atau permintaan ditolak "karena proxy penyimpanan kunci eksternal tidak merespons tepat waktu," itu mungkin menunjukkan bahwa manajer kunci eksternal Anda atau proxy penyimpanan kunci eksternal tidak dapat mengimbangi tingkat permintaan saat ini. Jika memungkinkan, turunkan tingkat permintaan Anda. Anda juga dapat mempertimbangkan untuk meminta penurunan nilai kuota permintaan toko kunci kustom Anda. Penurunan nilai kuota ini dapat meningkatkan pembatasan (dan nilai `ExternalKeyStoreThrottle` metrik), tetapi ini menunjukkan bahwa AWS KMS menolak permintaan berlebih dengan cepat sebelum dikirim ke proxy penyimpanan kunci eksternal atau pengelola kunci eksternal Anda. Untuk meminta pengurangan kuota, silakan kunjungi AWS Support Pusat dan buat kasus.

Nama grup dimensi: Keystore Throttle Metrics

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.
KmsOperation	Nilai untuk setiap operasi AWS KMS API. Metrik ini hanya berlaku untuk operasi kriptografi dan `DescribeKey` operasi pada kunci KMS di penyimpanan kunci eksternal.
KeySpec	Nilai untuk setiap jenis kunci KMS. Satu-satunya spesifikasi kunci yang didukung untuk kunci KMS di penyimpanan kunci eksternal adalah SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

Jumlah hari hingga sertifikat TLS untuk titik akhir proxy penyimpanan kunci eksternal Anda (`XksProxyUriEndpoint`) kedaluwarsa. Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

Gunakan metrik ini untuk membuat CloudWatch alarm yang memberi tahu Anda tentang kedaluwarsa sertifikat TLS Anda yang akan datang. Ketika sertifikat kedaluwarsa, AWS KMS tidak dapat

berkomunikasi dengan proxy penyimpanan kunci eksternal. Semua data yang dilindungi oleh kunci KMS di toko kunci eksternal Anda menjadi tidak dapat diakses sampai Anda memperbarui sertifikat.

Alarm sertifikat mencegah kedaluwarsa sertifikat yang mungkin mencegah Anda mengakses sumber daya terenkripsi Anda. Atur alarm untuk memberi waktu kepada organisasi Anda untuk memperbarui sertifikat sebelum kedaluwarsa.

Nama grup dimensi: Metrik Sertifikat Proxy XKS

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.
CertificateName	Nama subjek (CN) dalam sertifikat TLS.

XksProxyCredentialAge

Jumlah hari sejak kredensi proxy penyimpanan kunci eksternal saat ini (`XksProxyAuthenticationCredential`) dikaitkan dengan penyimpanan kunci eksternal. Hitungan ini dimulai ketika Anda memasukkan kredensi otentikasi sebagai bagian dari membuat atau memperbarui toko kunci eksternal Anda. Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

Nilai ini dirancang untuk mengingatkan Anda tentang usia kredensi otentikasi Anda. Namun, karena kami memulai penghitungan ketika Anda mengaitkan kredensi dengan penyimpanan kunci eksternal Anda, bukan saat Anda membuat kredensi otentikasi pada proxy penyimpanan kunci eksternal Anda, ini mungkin bukan indikator akurat usia kredensi pada proxy.

Gunakan metrik ini untuk membuat CloudWatch alarm yang mengingatkan Anda untuk memutar kredensi otentikasi proxy penyimpanan kunci eksternal Anda.

Nama grup dimensi: Metrik Per-Keystore

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.

XksProxyErrors

Jumlah pengecualian yang terkait dengan AWS KMS permintaan ke proxy penyimpanan kunci eksternal Anda. Jumlah ini mencakup pengecualian yang dikembalikan oleh proxy penyimpanan kunci eksternal AWS KMS dan kesalahan batas waktu yang terjadi ketika proxy penyimpanan kunci eksternal tidak merespons AWS KMS dalam interval waktu tunggu 250 milidetik. Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

Gunakan metrik ini untuk melacak tingkat kesalahan kunci KMS di toko kunci eksternal Anda. Ini mengungkapkan kesalahan yang paling sering, sehingga Anda dapat memprioritaskan upaya teknik Anda. Misalnya, kunci KMS yang menghasilkan tingkat kesalahan non-retryable yang tinggi mungkin menunjukkan masalah dengan konfigurasi penyimpanan kunci eksternal Anda. Untuk melihat konfigurasi penyimpanan kunci eksternal Anda, lihatMelihat toko kunci eksternal. Untuk mengedit pengaturan penyimpanan kunci eksternal Anda, lihatMengedit properti penyimpanan kunci eksternal.

Nama grup dimensi: Metrik Kesalahan Proksi XKS

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.
KmsOperation	Nilai untuk setiap operasi AWS KMS API yang menghasilkan permintaan ke proxy XKS.
XksOperation	Nilai untuk setiap operasi API proxy penyimpanan kunci eksternal.
KeySpec	Nilai untuk setiap jenis kunci KMS. Satu-satunya spesifikasi kunci yang didukung untuk kunci KMS di penyimpanan kunci eksternal adalah SYMMETRIC_DEFAULT.
ErrorType	Nilai: • Kesalahan yang dapat dicoba ulang: Kemungkinan bersifat sementara, seperti kesalahan jaringan. • Kesalahan yang tidak dapat dicoba ulang: Kemungkinan menunjukkan masalah dengan konfigurasi penyimpanan kunci khusus atau komponen eksternal. • N/A: Permintaan yang berhasil; tidak ada kesalahan

Dimensi	Deskripsi
Exception Name	Nilai: • Nama pengecualian • Tidak ada: Permintaan berhasil; tidak ada kesalahan

XksExternalKeyManagerStates

Hitungan jumlah instance manajer kunci eksternal di masing-masing status kesehatan berikut:`Active`,`Degraded`, dan`Unavailable`. Informasi untuk metrik ini berasal dari proxy penyimpanan kunci eksternal yang terkait dengan setiap penyimpanan kunci eksternal. Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

Berikut ini adalah status kesehatan untuk instance manajer kunci eksternal yang terkait dengan penyimpanan kunci eksternal. Setiap proxy penyimpanan kunci eksternal mungkin menggunakan indikator yang berbeda untuk mengukur status kesehatan manajer kunci eksternal Anda. Untuk detailnya, lihat dokumentasi untuk proxy penyimpanan kunci eksternal Anda.

- `Active`: Manajer kunci eksternal sehat.
- `Degraded`: Manajer kunci eksternal tidak sehat, tetapi masih dapat melayani lalu lintas
- `Unavailable`: Manajer kunci eksternal tidak dapat melayani lalu lintas.

Gunakan metrik ini untuk membuat CloudWatch alarm yang memberi tahu Anda tentang instans pengelola kunci eksternal yang terdegradasi dan tidak tersedia. Untuk menentukan instans pengelola kunci eksternal mana yang ada di setiap status, lihat log proxy penyimpanan kunci eksternal Anda.

Nama grup dimensi: Metrik Manajer Kunci Eksternal XKS

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.
XksExternalKeyManagerState	Nilai untuk setiap kondisi kesehatan.

XksProxyLatency

Jumlah milidetik yang diperlukan untuk proxy penyimpanan kunci eksternal untuk menanggapi AWS KMS permintaan. Jika waktu permintaan habis, nilai yang dicatat adalah batas waktu tunggu 250 milidetik. Metrik ini hanya berlaku untuk penyimpanan kunci eksternal.

Gunakan metrik ini untuk mengevaluasi kinerja proxy penyimpanan kunci eksternal dan pengelola kunci eksternal Anda. Misalnya, jika proxy sering kehabisan waktu pada operasi enkripsi dan dekripsi, konsultasikan dengan administrator proxy eksternal Anda.

Respons lambat mungkin juga menunjukkan bahwa pengelola kunci eksternal Anda tidak dapat menangani lalu lintas permintaan saat ini. AWS KMS merekomendasikan bahwa manajer kunci eksternal Anda dapat menangani hingga 1800 permintaan untuk operasi kriptografi per detik. Jika pengelola kunci eksternal Anda tidak dapat menangani tarif 1800 permintaan per detik, pertimbangkan untuk meminta penurunan kuota permintaan Anda untuk kunci KMS di toko kunci khusus. Permintaan untuk operasi kriptografi menggunakan kunci KMS di toko kunci eksternal Anda akan gagal dengan cepat dengan pengecualian pelambatan, daripada diproses dan kemudian ditolak oleh proxy penyimpanan kunci eksternal atau manajer kunci eksternal Anda.

Nama grup dimensi: Metrik Latensi Proxy XKS

Dimensi	Deskripsi
CustomKeyStoreId	Nilai untuk setiap penyimpanan kunci eksternal.
KmsOperation	Nilai untuk setiap operasi AWS KMS API yang menghasilkan permintaan ke proxy XKS.
XksOperation	Nilai untuk setiap operasi API proxy penyimpanan kunci eksternal.
KeySpec	Nilai untuk setiap jenis kunci KMS. Satu-satunya spesifikasi kunci yang didukung untuk kunci KMS di penyimpanan kunci eksternal adalah SYMMETRIC_DEFAULT.

Melihat AWS KMS metrik

Anda dapat melihat AWS KMS metrik menggunakan AWS Management Console dan Amazon CloudWatch API.

Untuk melihat metrik menggunakan konsol CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Jika perlu, ubah wilayahnya. Dari bilah navigasi, pilih wilayah tempat sumber daya AWS Anda berada.
3. Pada panel navigasi, silakan pilih Metrik, Semua metrik.
4. Pada tab Browse, cari`KMS`, dan mereka pilih KMS.
5. Pilih nama grup dimensi dari metrik yang ingin Anda lihat.

   Misalnya, untuk `SecondsUntilKeyMaterialExpiration` metrik, pilih Metrik Per-Kunci.
6. Untuk grafik nilai metrik, pilih nama metrik, lalu pilih`Add to graph`. Untuk mengonversi grafik garis menjadi nilai, pilih Garis, lalu pilih Angka.

Untuk melihat metrik menggunakan Amazon API CloudWatch

Untuk melihat AWS KMS metrik menggunakan CloudWatch API, kirim ListMetricspermintaan dengan `Namespace` set ke`AWS/KMS`. Contoh berikut ini menunjukkan cara melakukan dengan AWS Command Line Interface (AWS CLI).

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

Membuat CloudWatch alarm untuk memantau tombol KMS

Anda dapat membuat CloudWatch alarm Amazon berdasarkan AWS KMS metrik. Alarm mengirimkan pesan email ketika nilai metrik melebihi ambang batas yang ditentukan dalam konfigurasi alarm. Alarm dapat mengirim pesan email ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Auto Scaling Amazon EC2. Untuk informasi selengkapnya tentang CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon di Panduan Pengguna Amazon CloudWatch

Buat alarm untuk kedaluwarsa materi kunci impor

Anda dapat menggunakan `SecondsUntilKeyMaterialExpiration`metrik untuk membuat CloudWatch alarm yang memberi tahu Anda saat materi kunci yang diimpor dalam kunci KMS akan kedaluwarsa.

Saat Anda mengimpor materi kunci ke kunci KMS, Anda dapat secara opsional menentukan tanggal dan waktu saat materi kunci kedaluwarsa. Ketika materi kunci kedaluwarsa, AWS KMS menghapus materi kunci dan kunci KMS menjadi tidak dapat digunakan. Untuk menggunakan kunci KMS lagi, Anda harus mengimpor ulang materi kunci.

Untuk petunjuk, lihat Membuat CloudWatch alarm untuk kedaluwarsa materi kunci yang diimpor.

Buat alarm untuk penggunaan kunci KMS yang tertunda penghapusan

Saat Anda menjadwalkan penghapusan kunci KMS, AWS KMS memberlakukan masa tunggu sebelum menghapus kunci KMS. Anda dapat menggunakan masa tunggu untuk memastikan bahwa Anda tidak memerlukan kunci KMS sekarang atau di masa depan. Anda juga dapat mengonfigurasi

CloudWatch alarm untuk memperingatkan Anda jika seseorang atau aplikasi mencoba menggunakan kunci KMS dalam operasi kriptografi selama masa tunggu. Jika Anda menerima pemberitahuan dari alarm semacam itu, Anda mungkin ingin membatalkan penghapusan kunci KMS.

Untuk petunjuk, lihat Membuat alarm yang mendeteksi penggunaan kunci KMS tertunda penghapusan.

Buat alarm untuk memantau penyimpanan kunci eksternal

Anda dapat membuat CloudWatch alarm berdasarkan metrik untuk penyimpanan kunci eksternal dan kunci KMS di toko kunci eksternal.

Misalnya, kami menyarankan Anda menyetel CloudWatch alarm untuk memberi tahu Anda ketika sertifikat TLS untuk penyimpanan kunci eksternal Anda akan kedaluwarsa (XksProxyCertificateDaysToExpire), ketika proksi penyimpanan kunci eksternal Anda dan saat Anda melaporkan bahwa instance pengelola kunci eksternal Anda berada dalam status terdegradasi atau tidak tersedia (). XksProxyHsmStates

Untuk petunjuk, lihat Memantau toko kunci eksternal.

Pemantauan EventBridge dengan Amazon

Anda dapat menggunakan Amazon EventBridge (sebelumnya Amazon CloudWatch Events) untuk mengingatkan Anda tentang peristiwa penting berikut dalam siklus hidup kunci KMS Anda.

- Materi kunci dalam kunci KMS diputar secara otomatis.
- Materi kunci yang diimpor dalam kunci KMS kedaluwarsa.
- Kunci KMS yang telah dijadwalkan untuk dihapus telah dihapus.

AWS KMSterintegrasi dengan Amazon EventBridge untuk memberi tahu Anda tentang peristiwa penting yang memengaruhi kunci KMS Anda. Setiap peristiwa diwakili dalam JSON (JavaScriptObject Notation) dan termasuk nama acara, tanggal dan waktu ketika peristiwa terjadi, dan yang terpengaruh. Anda dapat mengumpulkan peristiwa ini dan menetapkan aturan yang merutekan mereka ke satu atau beberapa target seperti AWS Lambda fungsi, topik Amazon SNS, antrian Amazon SQS, streaming di Amazon Kinesis Data Streams, atau target bawaan.

Untuk informasi selengkapnya tentang penggunaan EventBridge dengan jenis peristiwa lain, termasuk peristiwa yang dipancarkan AWS CloudTrail saat merekam permintaan API baca/tulis, lihat Panduan Pengguna Amazon. EventBridge

Topik berikut menjelaskan EventBridge peristiwa yang AWS KMS dihasilkan.

Rotasi CMK KMS

AWS KMSmendukung rotasi otomatis bahan kunci dalam kunci KMS enkripsi simetris. Rotasi material kunci tahunan adalah opsional untuk kunci yang dikelola pelanggan. Bahan utama untuk Kunci yang dikelola AWSdiputar secara otomatis setiap tahun.

Setiap kali AWS KMS memutar materi kunci, ia mengirimkan `KMS CMK Rotation` acara ke EventBridge. AWS KMSmenghasilkan acara ini atas dasar upaya terbaik.

Berikut adalah contoh dari jenis peristiwa ini.

```
{
  "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"
  }
}
```

KMS Mengimpor Tanggal Kedaluwarsa Bahan Kunci

Saat Anda mengimpor materi kunci ke kunci KMS, Anda dapat secara opsional menentukan waktu di mana materi kunci kedaluwarsa. Ketika materi kunci kedaluwarsa, AWS KMS hapus materi kunci dan kirimkan `KMS Imported Key Material Expiration` acara yang sesuai ke. EventBridge AWS KMSmenghasilkan acara ini atas dasar upaya terbaik.

Berikut adalah contoh dari jenis peristiwa ini.

```
{
  "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"
  }
}
```

Penghapusan CMK KMS

Saat Anda menjadwalkan penghapusan kunci KMS, AWS KMS memberlakukan masa tunggu sebelum menghapus kunci KMS. Setelah masa tunggu berakhir, AWS KMS hapus kunci KMS dan kirim `KMS CMK Deletion` acara ke. EventBridge AWS KMSmenjamin EventBridge acara ini. Karena percobaan ulang, mungkin menghasilkan beberapa peristiwa dalam beberapa detik yang menghapus kunci KMS yang sama.

Berikut adalah contoh dari jenis peristiwa ini.

```
{
  "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"
  }
}
```

Menciptakan AWS KMS sumber daya dengan AWS CloudFormation

AWS Key Management Service terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat template yang menjelaskan kunci dan alias KMS, dan AWS CloudFormation ketentuan serta mengonfigurasi sumber daya tersebut untuk Anda. Untuk informasi tentang AWS KMS dukungan CloudFormation, lihat referensi jenis sumber daya KMS di Panduan AWS CloudFormation Pengguna.

Ketika Anda menggunakan AWS CloudFormation, Anda dapat menggunakan kembali template Anda untuk mengatur AWS KMS sumber daya Anda secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang di beberapa Akun AWS dan Wilayah.

Untuk menyediakan dan mengonfigurasi sumber daya untuk AWS KMS dan AWS layanan lainnya, Anda harus memahami AWS CloudFormation templat. Templat adalah file teks dengan format JSON atau YAML. Template ini menjelaskan sumber daya yang ingin Anda sediakan di AWS CloudFormation tumpukan Anda. Jika Anda tidak terbiasa dengan JSON atau YAMM, Anda dapat menggunakan AWS CloudFormation Designer untuk membantu Anda memulai dengan template. AWS CloudFormation Untuk informasi selengkapnya, lihat Apa itu AWS CloudFormation Designer? di Panduan Pengguna AWS CloudFormation .

Daerah

AWS KMS CloudFormation sumber daya didukung di semua Wilayah yang AWS CloudFormation didukung.

AWS KMS sumber daya dalam AWS CloudFormation template

AWS KMS mendukung AWS CloudFormation sumber daya berikut.

- Sumber `AWS::KMS::Key`daya menentukan kunci KMS di. AWS Key Management Service Anda dapat menggunakan sumber daya ini untuk membuat kunci KMS enkripsi simetris, kunci KMS asimetris untuk enkripsi atau penandatanganan, dan kunci KMS HMAC simetris. Anda dapat menggunakan `AWS::KMS::Key` untuk membuat kunci utama Multi-wilayah dari semua jenis yang didukung. Untuk mereplikasi kunci Multi-region, gunakan sumber daya. `AWS::KMS::ReplicaKey`
- `AWS::KMS::Alias`membuat alias dan mengaitkannya dengan kunci KMS. Kunci KMS dapat didefinisikan dalam template, atau dibuat oleh mekanisme lain.

- `AWS::KMS::ReplicaKey` membuat kunci replika multi-Wilayah. Untuk membuat bukti kunci primer multi-Wilayah, gunakan sumber daya `AWS::KMS::Key`. Anda tidak dapat menggunakan sumber daya ini untuk mereplikasi kunci Multi-wilayah dengan materi kunci yang diimpor. Untuk detail tentang kunci multi-Wilayah, lihat Kunci Multi-Region di AWS KMS.

> **Important**
>
> Jika Anda mengubah nilai`KeyUsage`,`KeySpec`, atau `MultiRegion` properti kunci KMS yang ada, kunci KMS yang ada dijadwalkan untuk dihapus dan kunci KMS baru dibuat dengan nilai yang ditentukan.
> Sementara dijadwalkan untuk dihapus, kunci KMS yang ada menjadi tidak dapat digunakan. Jika Anda tidak membatalkan penghapusan terjadwal dari kunci KMS yang ada di luar AWS CloudFormation, semua data yang dienkripsi di bawah kunci KMS yang ada menjadi tidak dapat dipulihkan ketika kunci KMS dihapus.

Kunci KMS yang dibuat template adalah sumber daya aktual di Anda Akun AWS. Prinsipal resmi dapat menggunakan dan mengelola kunci KMS yang dibuat template, baik dengan menggunakan template, AWS KMS konsol, atau API. AWS KMS Saat Anda menghapus kunci KMS dari template Anda, kunci KMS dijadwalkan untuk dihapus menggunakan masa tunggu yang Anda tentukan sebelumnya.

Misalnya, Anda dapat menggunakan AWS CloudFormation templat untuk membuat kunci KMS pengujian dengan kebijakan kunci, spesifikasi kunci, penggunaan kunci, alias, dan tag yang Anda inginkan. Anda dapat menjalankannya melalui serangkaian pengujian Anda, meninjau hasil Anda, lalu menggunakan templat untuk menjadwalkan kunci pengujian untuk penghapusan. Kemudian, Anda dapat menjalankan templat lagi untuk membuat kunci pengujian dengan properti yang sama.

Atau Anda dapat menggunakan AWS CloudFormation template untuk menentukan konfigurasi kunci KMS tertentu yang memenuhi aturan bisnis dan standar keamanan Anda. Kemudian Anda dapat menggunakan template itu kapan saja Anda perlu membuat kunci KMS. Anda tidak perlu khawatir tentang kunci yang salah konfigurasi. Jika konfigurasi pilihan Anda berubah, Anda dapat menggunakan template Anda untuk memperbarui kunci KMS Anda. Misalnya, template memudahkan untuk mengaktifkan rotasi tombol otomatis secara terprogram pada semua tombol KMS yang didefinisikan oleh template.

Untuk informasi selengkapnya tentang AWS KMS sumber daya, termasuk contoh, lihat referensi tipe sumber daya KMS di Panduan AWS CloudFormation Pengguna.

Pelajari lebih lanjut tentang AWS CloudFormation

Untuk mempelajari selengkapnya AWS CloudFormation, lihat sumber daya berikut:

- AWS CloudFormation
- AWS CloudFormation Panduan Pengguna
- AWS CloudFormation Referensi API
- Panduan Pengguna Antarmuka Baris Perintah AWS CloudFormation

Menghapus AWS KMS keys

Menghapus sebuah AWS KMS key adalah destruktif dan berpotensi berbahaya. Ini menghapus materi kunci dan semua metadata yang terkait dengan kunci KMS dan tidak dapat diubah. Setelah kunci KMS dihapus, Anda tidak dapat lagi mendekripsi data yang dienkripsi di bawah kunci KMS itu, yang berarti bahwa data menjadi tidak dapat dipulihkan. (Satu-satunya pengecualian adalah kunci replika multi-wilayah dan kunci KMS asimetris dan HMAC dengan bahan kunci yang diimpor.) Risiko ini signifikan untuk kunci KMS asimetris yang digunakan untuk enkripsi di mana, tanpa peringatan atau kesalahan, pengguna dapat terus menghasilkan ciphertext dengan kunci publik yang tidak dapat didekripsi setelah kunci pribadi dihapus. AWS KMS

Anda harus menghapus kunci KMS hanya ketika Anda yakin bahwa Anda tidak perlu menggunakannya lagi. Jika Anda tidak yakin, pertimbangkan untuk menonaktifkan kunci KMS alih-alih menghapusnya. Anda dapat mengaktifkan kembali kunci KMS yang dinonaktifkan dan membatalkan penghapusan kunci KMS yang dijadwalkan, tetapi Anda tidak dapat memulihkan kunci KMS yang dihapus.

Anda hanya dapat menjadwalkan penghapusan kunci yang dikelola pelanggan. Anda tidak dapat menghapus Kunci yang dikelola AWS atauKunci milik AWS.

Sebelum menghapus kunci KMS, Anda mungkin ingin tahu berapa banyak ciphertext yang dienkripsi di bawah kunci KMS itu. AWS KMStidak menyimpan informasi ini dan tidak menyimpan ciphertext apa pun. Untuk mendapatkan informasi ini, Anda harus menentukan penggunaan kunci KMS sebelumnya. Untuk bantuan, pergi keMenentukan penggunaan kunci KMS di masa lalu.

AWS KMStidak pernah menghapus kunci KMS Anda kecuali Anda secara eksplisit menjadwalkannya untuk dihapus dan masa tunggu wajib berakhir.

Namun, Anda dapat memilih untuk menghapus kunci KMS karena satu atau beberapa alasan berikut:

- Untuk menyelesaikan siklus hidup kunci untuk kunci KMS yang tidak lagi Anda perlukan
- Untuk menghindari overhead manajemen dan biaya yang terkait dengan pemeliharaan kunci KMS yang tidak digunakan
- Untuk mengurangi jumlah kunci KMS yang dihitung terhadap kuota sumber daya kunci KMS Anda

> **Note**
>
> Jika Anda menutup Akun AWS, kunci KMS Anda menjadi tidak dapat diakses dan Anda tidak lagi ditagih untuk itu.

AWS KMSmencatat entri di AWS CloudTrail log Anda ketika Anda menjadwalkan penghapusan kunci KMS dan ketika kunci KMS benar-benar dihapus.

Untuk mengetahui informasi tentang menghapus multi-Wilayah utama dan replika kunci, lihat Menghapus kunci multi-Wilayah.

Topik

Tentang masa tunggu

Karena merusak dan berpotensi berbahaya untuk menghapus kunci KMS, AWS KMS mengharuskan Anda untuk menetapkan masa tunggu 7 - 30 hari. Masa tunggu default adalah 30 hari.

Namun, masa tunggu sebenarnya mungkin hingga 24 jam lebih lama dari yang Anda jadwalkan. Untuk mendapatkan tanggal dan waktu aktual ketika kunci KMS akan dihapus, gunakan

DescribeKeyoperasi. Atau di AWS KMS konsol, pada halaman detail untuk kunci KMS, di bagian Konfigurasi umum, lihat Tanggal penghapusan terjadwal. Pastikan untuk mencatat zona waktu.

Selama masa tunggu, status kunci KMS dan status kunci adalah Penghapusan tertunda.

- Penghapusan kunci KMS yang tertunda tidak dapat digunakan dalam operasi kriptografi apa pun.
- AWS KMStidak memutar bahan kunci kunci KMS yang menunggu penghapusan.

Setelah masa tunggu berakhir, AWS KMS hapus kunci KMS, aliasnya, dan semua metadata terkait. AWS KMS

Penjadwalan penghapusan kunci KMS mungkin tidak langsung mempengaruhi kunci data yang dienkripsi oleh kunci KMS. Untuk detailnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Gunakan masa tunggu untuk memastikan bahwa Anda tidak memerlukan kunci KMS sekarang atau di masa depan. Anda dapat mengonfigurasi CloudWatch alarm Amazon untuk memperingatkan Anda jika seseorang atau aplikasi mencoba menggunakan kunci KMS selama masa tunggu. Untuk memulihkan kunci KMS, Anda dapat membatalkan penghapusan kunci sebelum masa tunggu berakhir. Setelah masa tunggu berakhir, Anda tidak dapat membatalkan penghapusan kunci, dan AWS KMS menghapus kunci KMS.

Menghapus tombol KMS asimetris

Pengguna yang berwenang dapat menghapus kunci KMS simetris atau asimetris. Prosedur untuk menjadwalkan penghapusan kunci KMS ini sama untuk kedua jenis kunci. Namun, karena kunci publik dari kunci KMS asimetris dapat diunduh dan digunakan di luar, operasi menimbulkan risiko tambahan yang signifikanAWS KMS, terutama untuk kunci KMS asimetris yang digunakan untuk enkripsi (penggunaan kuncinya adalah). `ENCRYPT_DECRYPT`

- Saat Anda menjadwalkan penghapusan kunci KMS, status kunci kunci KMS berubah menjadi penghapusan Tertunda, dan kunci KMS tidak dapat digunakan dalam operasi kriptografi. Namun, penjadwalan penghapusan tidak berpengaruh pada kunci publik di luar AWS KMS. Pengguna yang memiliki kunci publik dapat terus menggunakannya untuk mengenkripsi pesan. Mereka tidak menerima notifikasi bahwa status kunci berubah. Kecuali penghapusan dibatalkan, ciphertext yang dibuat dengan kunci publik tidak dapat didekripsi.
- Alarm, log, dan strategi lain yang mendeteksi upaya penggunaan kunci KMS yang tertunda penghapusan tidak dapat mendeteksi penggunaan kunci publik di luar. AWS KMS

- Ketika kunci KMS dihapus, semua AWS KMS tindakan yang melibatkan kunci KMS gagal. Namun, pengguna yang memiliki kunci publik dapat terus menggunakannya untuk mengenkripsi pesan. Ciphertext ini tidak dapat didekripsi.

Jika Anda harus menghapus kunci KMS asimetris dengan penggunaan kunci`ENCRYPT_DECRYPT`, gunakan entri CloudTrail Log Anda untuk menentukan apakah kunci publik telah diunduh dan dibagikan. Jika memiliki, lakukan verifikasi bahwa kunci publik tidak sedang digunakan di luar AWS KMS. Kemudian, pertimbangkan untuk menonaktifkan kunci KMS alih-alih menghapusnya.

Risiko yang ditimbulkan dengan menghapus kunci KMS asimetris dikurangi untuk kunci KMS asimetris dengan bahan kunci yang diimpor. Untuk detailnya, lihat Menghapus kunci KMS dengan materi kunci yang diimpor.

Menghapus kunci multi-Wilayah

Pengguna yang diotorisasi dapat menjadwalkan penghapusan multi-Wilayah utama dan kunci replika. Namun, AWS KMS tidak akan menghapus bukti kunci utama multi-Wilayah yang memiliki kunci replika. Demikian juga, selama kunci utama ada, Anda dapat membuat ulang kunci replika multi-Wilayah yang dihapus. Untuk mengetahui detail selengkapnya, lihat Menghapus kunci multi-Wilayah.

Menghapus kunci KMS dengan bahan kunci yang diimpor

Pengguna yang berwenang dapat menjadwalkan penghapusan kunci KMS dengan bahan kunci yang diimpor. Tindakan ini secara permanen menghapus kunci KMS, materi utamanya, dan semua metadata yang terkait dengan kunci KMS.

Anda tidak dapat membuat kunci KMS enkripsi simetris baru yang dapat mendekripsi ciphertext dari kunci enkripsi simetris yang dihapus dengan materi kunci yang diimpor, bahkan jika Anda memiliki salinan materi utamanya. Namun, jika Anda memiliki materi utama, Anda dapat secara efektif membuat ulang kunci KMS asimetris atau kunci KMS HMAC dengan bahan kunci impor. Untuk detailnya, lihat Menghapus kunci KMS dengan materi kunci yang diimpor.

Mengontrol akses ke penghapusan kunci

Jika Anda menggunakan kebijakan IAM untuk mengizinkan AWS KMS izin, identitas IAM yang memiliki akses AWS administrator (`"Action": "*"`) atau akses AWS KMS penuh () sudah diizinkan untuk menjadwalkan dan membatalkan kunci penghapusan kunci KMS. `"Action": "kms:*"` Untuk mengizinkan administrator kunci menjadwalkan dan membatalkan penghapusan kunci dalam kebijakan kunci, gunakan AWS KMS konsol atau API. AWS KMS

Biasanya, hanya administrator kunci yang memiliki izin untuk menjadwalkan atau membatalkan penghapusan kunci. Namun, Anda dapat memberikan izin ini ke identitas IAM lainnya dengan menambahkan `kms:CancelKeyDeletion` izin `kms:ScheduleKeyDeletion` dan ke kebijakan kunci atau kebijakan IAM. Anda juga dapat menggunakan kunci `kms:ScheduleKeyDeletionPendingWindowInDays`kondisi untuk lebih membatasi nilai yang dapat ditentukan oleh prinsipal dalam `PendingWindowInDays` parameter permintaan. ScheduleKeyDeletion

Izinkan administrator kunci untuk menjadwalkan dan membatalkan penghapusan kunci (konsol)

Untuk memberikan izin kepada administrator kunci untuk menjadwalkan dan membatalkan penghapusan kunci.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih alias atau ID kunci kunci KMS yang izinnya ingin Anda ubah.
5. Pilih tab kebijakan kunci.
6. Langkah selanjutnya berbeda untuk tampilan default dan tampilan kebijakan kebijakan utama Anda. Tampilan default hanya tersedia jika Anda menggunakan kebijakan kunci konsol default. Jika tidak, hanya tampilan kebijakan yang tersedia.

   Jika tampilan default tersedia, tombol Beralih ke tampilan kebijakan atau Beralih ke tampilan default akan muncul di tab Kebijakan kunci.

   - Dalam tampilan default:
     - Di bawah Penghapusan kunci, pilih Izinkan administrator kunci untuk menghapus kunci ini.
   - Dalam tampilan kebijakan:
     a. Pilih Edit.
     b. Dalam pernyataan kebijakan untuk administrator kunci, tambahkan `kms:CancelKeyDeletion` izin `kms:ScheduleKeyDeletion` dan ke elemen. `Action`

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

c. Pilih Simpan perubahan.

Izinkan izin administrator kunci untuk menjadwalkan dan membatalkan penghapusan kunci () AWS CLI

Anda dapat menggunakan AWS Command Line Interface untuk menambahkan izin menjadwalkan dan membatalkan penghapusan kunci.

Guna menambahkan izin untuk menjadwalkan dan membatalkan penghapusan kunci

1. Gunakan perintah `aws kms get-key-policy` untuk mengambil kebijakan kunci yang sudah ada, lalu simpan dokumen kebijakan ke sebuah file.
2. Buka dokumen kebijakan di editor teks pilihan Anda. Dalam pernyataan kebijakan untuk administrator kunci, tambahkan `kms:ScheduleKeyDeletion` dan `kms:CancelKeyDeletion` izin. Contoh berikut menunjukkan pernyataan kebijakan dengan dua izin ini:

```
{
  "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. Gunakan `aws kms put-key-policy`perintah untuk menerapkan kebijakan kunci ke tombol KMS.

Menjadwalkan dan membatalkan penghapusan kunci

Prosedur berikut menjelaskan cara menjadwalkan penghapusan kunci dan membatalkan penghapusan kunci wilayah Tunggal AWS KMS keys (kunci KMS) dalam AWS KMS menggunakan,, danAWS Management Console. AWS CLI AWS SDK for Java

Untuk informasi tentang penjadwalan penghapusan tombol multi-Wilayah, lihat Menghapus kunci multi-Wilayah.

> **Warning**
>
> Menghapus kunci KMS bersifat merusak dan berpotensi berbahaya. Anda harus melanjutkan hanya ketika Anda yakin bahwa Anda tidak perlu menggunakan kunci KMS lagi dan tidak perlu menggunakannya di masa depan. Jika Anda tidak yakin, Anda harus menonaktifkan kunci KMS alih-alih menghapusnya.

Sebelum Anda dapat menghapus kunci KMS, Anda harus memiliki izin untuk melakukannya. Untuk informasi tentang memberikan izin ini kepada administrator utama, lihat. Mengontrol akses ke penghapusan kunci Anda juga dapat menggunakan kunci

`kms:ScheduleKeyDeletionPendingWindowInDays`kondisi untuk lebih membatasi masa tunggu, seperti menegakkan masa tunggu minimum.

AWS KMSmencatat entri di AWS CloudTrail log Anda ketika Anda menjadwalkan penghapusan kunci KMS dan ketika kunci KMS benar-benar dihapus.

Menjadwalkan dan membatalkan penghapusan kunci (konsol)

DiAWS Management Console, Anda dapat menjadwalkan dan membatalkan penghapusan beberapa kunci KMS sekaligus.

Untuk menjadwalkan penghapusan kunci

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.

   Anda tidak dapat menjadwalkan penghapusan atau. Kunci yang dikelola AWSKunci milik AWS
4. Pilih kotak centang di sebelah tombol KMS yang ingin Anda hapus.
5. Pilih Tindakan kunci, Jadwalkan penghapusan kunci.
6. Baca dan pertimbangkan peringatan, dan informasi tentang membatalkan penghapusan selama masa tunggu. Jika Anda memutuskan untuk membatalkan penghapusan, di bagian bawah halaman, pilih Batalkan.
7. Untuk Masa tunggu (dalam hari), masukkan beberapa hari antara 7 dan 30.
8. Tinjau kunci KMS yang Anda hapus.
9. <number of days>Pilih kotak centang di samping Konfirmasi bahwa Anda ingin menjadwalkan kunci ini untuk dihapus dalam beberapa hari. .
10. Pilih Jadwalkan penghapusan.

Status kunci KMS berubah menjadi penghapusan Tertunda.

Untuk membatalkan penghapusan kunci

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilihan Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.

4. Pilih kotak centang di sebelah tombol KMS yang ingin Anda pulihkan.
5. Pilih Tindakan kunci,Batalkan penghapusan kunci.

Status kunci KMS berubah dari penghapusan Tertunda ke Dinonaktifkan. Untuk menggunakan kunci KMS, Anda harus mengaktifkannya.

Menjadwalkan dan membatalkan penghapusan kunci (AWS CLI)

Gunakan `aws kms schedule-key-deletion`perintah untuk menjadwalkan penghapusan kunci dari kunci yang dikelola pelanggan, seperti yang ditunjukkan pada contoh berikut.

Anda tidak dapat menjadwalkan penghapusan atauKunci yang dikelola AWS. Kunci milik AWS

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --
pending-window-in-days 10
```

Ketika berhasil digunakan, AWS CLI mengembalikan output seperti output yang ditunjukkan dalam contoh berikut:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": 1598304792.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 10
}
```

Gunakan perintah `aws kms cancel-key-deletion` untuk menjadwalkan penghapusan kunci dari AWS CLI seperti yang ditunjukkan dalam contoh berikut.

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

Ketika berhasil digunakan, AWS CLI mengembalikan output seperti output yang ditunjukkan dalam contoh berikut:

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

Status kunci KMS berubah dari Penghapusan Tertunda ke Dinonaktifkan. Untuk menggunakan kunci KMS, Anda harus mengaktifkannya.

Menjadwalkan dan membatalkan penghapusan kunci (AWS SDK for Java)

Contoh berikut menunjukkan bagaimana menjadwalkan penghapusan kunci yang dikelola pelanggan dengan. AWS SDK for Java Contoh ini mengharuskan Anda sebelumnya menginstankan `AWSKMSClient` sebagai `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);
```

Contoh berikut menunjukkan cara membatalkan penghapusan kunci dengan AWS SDK for Java. Contoh ini mengharuskan Anda sebelumnya menginstankan `AWSKMSClient` sebagai `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);
```

Status kunci KMS berubah dari Penghapusan Tertunda ke Dinonaktifkan. Untuk menggunakan kunci KMS, Anda harus mengaktifkannya.

Membuat alarm yang mendeteksi penggunaan kunci KMS tertunda penghapusan

Anda dapat menggabungkan fiturAWS CloudTrail, Amazon CloudWatch Logs, dan Amazon Simple Notification Service (Amazon SNS) untuk membuat alarm CloudWatch Amazon yang memberi tahu Anda ketika seseorang di akun Anda mencoba menggunakan kunci KMS yang sedang menunggu penghapusan. Jika Anda menerima pemberitahuan ini, Anda mungkin ingin membatalkan penghapusan kunci KMS dan mempertimbangkan kembali keputusan Anda untuk menghapusnya.

Prosedur berikut membuat alarm yang memberi tahu Anda setiap kali pesan kesalahan *`Key ARN`* `is pending deletion` "" ditulis ke file CloudTrail log Anda. Pesan kesalahan ini menunjukkan bahwa seseorang atau aplikasi mencoba menggunakan kunci KMS dalam operasi kriptografi. Karena notifikasi ditautkan ke pesan kesalahan, notifikasi tidak dipicu saat Anda menggunakan operasi API yang diizinkan pada kunci KMS yang menunggu penghapusan, sepertiListKeys,, dan. CancelKeyDeletion PutKeyPolicy Untuk melihat daftar operasi API AWS KMS yang mengembalikan pesan kesalahan ini, lihat Status AWS KMS kunci kunci.

Email notifikasi yang Anda terima tidak mencantumkan kunci KMS atau operasi kriptografi. Anda dapat menemukan informasi itu di CloudTraillog Anda. Sebaliknya, email melaporkan bahwa status alarm berubah dari OKE ke Alarm. Untuk informasi selengkapnya tentang CloudWatch alarm dan perubahan status, lihat Menggunakan CloudWatch alarm Amazon di CloudWatch Panduan Pengguna Amazon.

> **Warning**
>
> CloudWatch Alarm Amazon ini tidak dapat mendeteksi penggunaan kunci publik dari kunci KMS asimetris di luar. AWS KMS Untuk detail tentang risiko khusus menghapus kunci KMS asimetris yang digunakan untuk kriptografi kunci publik, termasuk membuat ciphertext yang tidak dapat didekripsi, lihat. Menghapus tombol KMS asimetris

Topik

- Persyaratan untuk CloudWatch alarm
- Membuat CloudWatch alarm

Persyaratan untuk CloudWatch alarm

Sebelum membuat CloudWatch alarm, Anda harus membuat AWS CloudTrail jejak dan mengonfigurasi CloudTrail untuk mengirimkan file CloudTrail log ke Amazon CloudWatch Logs. Anda juga memerlukan topik Amazon SNS untuk pemberitahuan alarm.

- Buat CloudTrail jejak.

  CloudTrail secara otomatis diaktifkan pada Anda Akun AWS ketika Anda membuat akun. Namun, untuk catatan berkelanjutan tentang peristiwa di akun Anda, termasuk peristiwa untuk AWS KMS, buat jejak.

- Konfigurasikan CloudTrail untuk mengirimkan file log Anda CloudWatch Log.

  Konfigurasikan pengiriman file CloudTrail log Anda ke CloudWatch Log. Hal ini memungkinkan CloudWatch Log untuk memantau log untuk permintaan AWS KMS API yang mencoba menggunakan kunci KMS yang sedang menunggu penghapusan.

- Buat topik Amazon SNS.

  Saat alarm Anda terpicu, alarm akan memberi tahu Anda dengan mengirim pesan email ke alamat email dalam topik Amazon Simple Notification Service (Amazon SNS).

Membuat CloudWatch alarm

Dalam prosedur ini, Anda membuat filter metrik grup CloudWatch log yang menemukan contoh pengecualian penghapusan tertunda. Kemudian, Anda membuat CloudWatch alarm berdasarkan metrik grup log. Untuk informasi tentang filter metrik grup log, lihat Membuat metrik dari peristiwa log menggunakan filter di Panduan Pengguna CloudWatch Log Amazon.

1. Buat filter CloudWatch metrik yang mem-parsing CloudTrail log.

   Ikuti petunjuk di Buat filter metrik untuk grup log menggunakan nilai wajib berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pola filter	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
Nilai metrik	1

2. Buat CloudWatch alarm berdasarkan filter metrik yang Anda buat di Langkah 1.

   Ikuti petunjuk di Membuat CloudWatch alarm berdasarkan filter metrik grup log menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Filter metrik	Nama filter metrik yang Anda buat di Langkah 1.

Bidang	Nilai
Jenis ambang	Statis
Kondisi	Setiap kali ***nama metrik lebih*** besar dari 1
Datapoint untuk alarm	1keluar dari 1
Perlakuan data yang hilang	Perlakukan data yang hilang sebagai hal yang baik (tidak melanggar ambang batas)

Setelah Anda menyelesaikan prosedur ini, Anda akan menerima pemberitahuan setiap kali CloudWatch alarm baru Anda memasuki `ALARM` status. Jika Anda menerima pemberitahuan untuk alarm ini, itu mungkin berarti bahwa kunci KMS yang dijadwalkan untuk dihapus masih diperlukan untuk mengenkripsi atau mendekripsi data. Dalam hal ini, batalkan penghapusan kunci KMS dan pertimbangkan kembali keputusan Anda untuk menghapusnya.

Menentukan penggunaan kunci KMS di masa lalu

Sebelum menghapus kunci KMS, Anda mungkin ingin tahu berapa banyak ciphertext yang dienkripsi di bawah kunci itu. AWS KMStidak menyimpan informasi ini, dan tidak menyimpan ciphertext apa pun. Mengetahui bagaimana kunci KMS digunakan di masa lalu dapat membantu Anda memutuskan apakah Anda akan membutuhkannya di masa depan atau tidak. Topik ini menyarankan beberapa strategi yang dapat membantu Anda menentukan penggunaan kunci KMS di masa lalu.

> **Warning**
>
> Strategi ini untuk menentukan penggunaan masa lalu dan aktual hanya efektif bagi pengguna AWS dan operasi AWS KMS. Mereka tidak dapat mendeteksi penggunaan kunci publik dari kunci KMS asimetris di luar. AWS KMS Untuk detail tentang risiko khusus menghapus kunci KMS asimetris yang digunakan untuk kriptografi kunci publik, termasuk membuat ciphertext yang tidak dapat didekripsi, lihat. Menghapus tombol KMS asimetris

Topik

- Memeriksa izin kunci KMS untuk menentukan ruang lingkup penggunaan potensial

- Memeriksa log AWS CloudTrail untuk menentukan penggunaan aktual

Memeriksa izin kunci KMS untuk menentukan ruang lingkup penggunaan potensial

Menentukan siapa atau apa yang saat ini memiliki akses ke kunci KMS dapat membantu Anda menentukan seberapa luas kunci KMS digunakan dan apakah masih diperlukan. Untuk mempelajari cara menentukan siapa atau apa yang saat ini memiliki akses ke kunci KMS, buka. Menentukan akses ke AWS KMS keys

Memeriksa log AWS CloudTrail untuk menentukan penggunaan aktual

Anda mungkin dapat menggunakan riwayat penggunaan kunci KMS untuk membantu Anda menentukan apakah Anda memiliki ciphertext yang dienkripsi di bawah kunci KMS tertentu.

Semua aktivitas API AWS KMS dicatat dalam file log AWS CloudTrail. Jika Anda telah membuat CloudTrail jejak di wilayah tempat kunci KMS berada, Anda dapat memeriksa file CloudTrail log untuk melihat riwayat semua aktivitas AWS KMS API untuk kunci KMS tertentu. Jika Anda tidak memiliki jejak, Anda masih dapat melihat peristiwa terbaru dalam riwayat CloudTrail acara Anda. Untuk detail tentang cara AWS KMS penggunaan CloudTrail, lihatLogging panggilan AWS KMS API dengan AWS CloudTrail.

Contoh berikut menunjukkan entri CloudTrail log yang dihasilkan saat kunci KMS digunakan untuk melindungi objek yang disimpan di Amazon Simple Storage Service (Amazon S3). Dalam contoh ini, objek diunggah ke Amazon S3 menggunakan Melindungi data menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS). Saat Anda mengunggah objek ke Amazon S3 dengan SSE-KMS, Anda menentukan kunci KMS yang akan digunakan untuk melindungi objek. Amazon S3 menggunakan AWS KMS GenerateDataKeyoperasi untuk meminta kunci data unik untuk objek, dan peristiwa permintaan ini masuk CloudTrail dengan entri yang mirip dengan berikut ini:

```
{
  "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"
}
```

Saat nanti Anda mengunduh objek ini dari Amazon S3, Amazon S3 mengirimkan permintaan AWS KMS untuk mendekripsi kunci data objek `Decrypt` menggunakan kunci KMS yang ditentukan. Ketika Anda melakukan ini, file CloudTrail log Anda menyertakan entri yang mirip dengan berikut ini:

```
{
  "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"
}
```

Semua aktivitas AWS KMS API dicatat oleh CloudTrail. Dengan mengevaluasi entri log ini, Anda mungkin dapat menentukan penggunaan sebelumnya dari kunci KMS tertentu, dan ini dapat membantu Anda menentukan apakah Anda ingin menghapusnya atau tidak.

Untuk melihat lebih banyak contoh bagaimana aktivitas AWS KMS API muncul di file CloudTrail log Anda, bukaLogging panggilan AWS KMS API dengan AWS CloudTrail. Untuk informasi lebih lanjut tentang CloudTrail pergi ke Panduan AWS CloudTrail Pengguna.

Status AWS KMS kunci kunci

AWS KMS key Selalu memiliki status kunci. Operasi pada kunci KMS dan lingkungannya dapat mengubah status kunci itu, baik sementara, atau sampai operasi lain mengubah status kuncinya.

Tabel di bagian ini menunjukkan bagaimana status kunci memengaruhi panggilan ke operasi AWS KMS API. Sebagai hasil dari status kuncinya, operasi pada kunci KMS diharapkan berhasil (#), gagal (X), atau berhasil hanya dalam kondisi tertentu (?). Hasilnya sering berbeda untuk kunci KMS dengan bahan kunci impor.

Tabel ini hanya mencakup operasi API yang menggunakan kunci KMS yang ada. Operasi lain, seperti CreateKeydan ListKeys, dihilangkan.

Topik

- Status kunci dan tipe kunci KMS
- Tabel status kunci

Status kunci dan tipe kunci KMS

Jenis kunci KMS menentukan status kunci yang dapat dimilikinya.

- Semua kunci KMS dapat berada di`Enabled`,`Disabled`, dan `PendingDeletion` negara bagian.
- Sebagian besar kunci KMS dibuat di `Enabled` negara bagian. Kunci dengan material kunci yang diimpor dibuat di status `PendingImport`.
- `PendingImport`Negara hanya berlaku untuk kunci KMS dengan bahan kunci yang diimpor.
- `Unavailable`Status hanya berlaku untuk kunci KMS di toko kunci khusus. Kunci KMS di toko AWS CloudHSM kunci adalah `Unavailable` ketika toko kunci khusus sengaja terputus dari klasternya. AWS CloudHSM Kunci KMS di penyimpanan kunci eksternal adalah `Unavailable` ketika toko kunci kustom sengaja terputus dari proxy penyimpanan kunci eksternal. Anda dapat

melihat dan mengelola kunci KMS yang tidak tersedia, tetapi Anda tidak dapat menggunakannya dalam operasi kriptografi.

Status kunci kunci KMS di toko kunci khusus tidak terpengaruh oleh perubahan pada kunci dukungannya. Kunci KMS di toko AWS CloudHSM kunci tidak terpengaruh oleh perubahan pada materi kunci terkait di AWS CloudHSM cluster. Kunci KMS di penyimpanan kunci eksternal tidak terpengaruh oleh perubahan pada kunci eksternal di manajer kunci eksternal. Jika kunci dukungan dinonaktifkan atau dihapus, status kunci KMS tidak berubah, tetapi operasi kriptografi menggunakan kunci KMS gagal.

- Status kunci `Creating`, `Updating`, dan `PendingReplicaDeletion` hanya berlaku untuk kunci multi-wilayah.
  - Kunci replika multi-wilayah adalah status kunci `Creating` sementara saat sedang dibuat. Proses ini mungkin masih berlangsung ketika ReplicateKeyoperasi selesai. Ketika proses replikasi selesai, kunci replika dalam status `Enabled` atau `PendingImport`.
  - Kunci multi-Wilayah berada dalam status kunci `Updating` sementara saat Wilayah utama sedang diperbarui. Proses ini mungkin masih berlangsung ketika UpdatePrimaryRegionoperasi selesai. Ketika proses pembaruan selesai, kunci primer dan replika melanjutkan status kunci `Enabled`.
  - Saat Anda menjadwalkan penghapusan kunci primer multi-Wilayah yang memiliki kunci replika, kunci utama berada dalam status `PendingReplicaDeletion` hingga semua kunci replikanya dihapus. Kemudian status kunci berubah menjadi `PendingDeletion`. Untuk rincian selengkapnya, lihat Menghapus kunci multi-Wilayah.

Tabel status kunci

Tabel berikut menunjukkan bagaimana keadaan kunci kunci KMS mempengaruhi AWS KMS operasi.

Deskripsi catatan kaki bernomor ([n]) berada di akhir topik ini.

> **Note**
>
> Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
CancelKeyDeletion	✗ [4]	✗ [4]	✓	✗ [4]	✗ [4], [13]	✗ [4]	✗ [4]
CreateAlias	✓	✓	✗ [3]	✓	✓	✓	✓
CreateGrant	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✓	✗ [14]	✓
Dekripsi	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
DeleteAlias	✓	✓	✓	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓ (Tidak ada efek)	N/A	✗ [14]	✗ [15]

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
DisableKeyRotation	? [7]	✗ [1] atau [7]	✗ [3] atau [7]	✗ [6]	✗ [7]	✗ [14]	? [7]
EnableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
EnableKeyRotation	? [7]	✗ [1] atau [7]	✗ [3] atau [7]	✗ [6]	✗ [7]	✗ [14]	? [7]
Enkripsi	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
GenerateDataKey	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateDataKeyPair	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	✗ [1]	✗ [2] atau [3]	✗ [5]	✗ [11]	✗ [14]	✓
GenerateMac	✓	✗ [1]	✗ [2] atau [3]	N/A	N/A	✗ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	? [7]	? [7]	? [7]	⊘ [6]	⊘ [7]	? [7]	? [7]
GetParametersForImport	? [9]	? [9]	⊘ [8] atau [9]	✓	⊘ [9]	⊘ [14]	⊘ [15]
GetPublicKey	✓	⊘ [1]	⊘ [2] atau [3]	N/A	N/A	⊘ [14]	✓
ImportKeyMaterial	? [9]	? [9]	⊘ [8] atau [9]	✓	⊘ [9]	⊘ [14]	✓
ListAliases	✓	✓	✓	✓	✓	✓	✓

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
ListGrants	✓	✓	✓	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓	✓	✓	✓
ListKeyRotations	? [7]	? [7]	? [7]	⊗ [6]	⊗ [7]	? [7]	? [7]
ListResourceTags	✓	✓	✓	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
ReplicateKey	✓	⊗ [1]	⊗ [2] atau [3]	⊗ [5]	N/A	⊗ [14]	⊗ [15]

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
RotateKeyOnDemand	? [7]	⊗ [1] atau [7]	⊗ [3] atau [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]
ScheduleKeyDeletion	✓	✓	⊗ [3]	✓	✓	✓	⊗ [15]
Sign	✓	⊗ [1]	⊗ [2] atau [3]	N/A	N/A	⊗ [14]	✓
TagResource	✓	✓	⊗ [3]	✓	✓	✓	✓

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
UntagResource	✓	✓	⊘ [3]	✓	✓	✓	✓
UpdateAlias	✓	✓	? [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	⊘ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊘ [1]	⊘ [2] atau [3]	⊘ [5]	N/A	⊘ [14]	✓
Verifikasi	✓	⊘ [1]	⊘ [2] atau [3]	N/A	N/A	⊘ [14]	✓

API	Diaktifkan	Nonaktif	Penghapusan yang tertunda Penghapusan replika yang tertunda	Menunggu impor	Tidak tersedia	Membuat	Memperbarui
VerifyMac	✓	✗ [1]	✗ [2] atau [3]	N/A	N/A	✗ [14]	✓

Rincian Tabel

- [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] Jika kunci KMS telah mengimpor bahan kunci atau ada di toko kunci khusus:`UnsupportedOperationException`.
- [8] Jika kunci KMS telah mengimpor bahan kunci: `KMSInvalidStateException`
- [9] Jika kunci KMS tidak dapat atau tidak memiliki materi kunci impor:`UnsupportedOperationException`.

- [10] Jika kunci KMS sumber tertunda penghapusan, perintah berhasil. Jika kunci KMS tujuan tertunda penghapusan, perintah gagal dengan kesalahan: `KMSInvalidStateException : <key ARN> is pending deletion.`
- [11] `KMSInvalidStateException: <key ARN> is unavailable.` Anda tidak dapat melakukan operasi ini pada kunci KMS yang tidak tersedia.
- [12] Operasi berhasil, tetapi status kunci dari kunci KMS tidak berubah sampai tersedia.
- [13] Sementara kunci KMS di toko kunci khusus sedang menunggu penghapusan, status kuncinya tetap ada `PendingDeletion` meskipun kunci KMS menjadi tidak tersedia. Ini memungkinkan Anda untuk membatalkan penghapusan kunci KMS kapan saja selama masa tunggu.
- [14] `KMSInvalidStateException: <key ARN> is creating.` AWS KMS melempar pengecualian ini saat mereplikasi kunci Multi-region (). `ReplicateKey`
- [15] `KMSInvalidStateException: <key ARN> is updating.` AWS KMS melempar pengecualian ini saat memperbarui Wilayah utama dari kunci Multi-wilayah ()`UpdatePrimaryRegion`.

Kontrol autentikasi dan akses untuk AWS KMS

Untuk menggunakannyaAWS KMS, Anda harus memiliki kredensyal yang AWS dapat digunakan untuk mengautentikasi permintaan Anda. Kredensyal harus menyertakan izin untuk mengakses AWS sumber daya: AWS KMS keysdan alias. Tidak ada AWS prinsipal yang memiliki izin untuk kunci KMS kecuali izin tersebut diberikan secara eksplisit dan tidak pernah ditolak. Tidak ada izin implisit atau otomatis untuk menggunakan atau mengelola kunci KMS.

Cara utama untuk mengelola akses ke AWS KMS sumber daya Anda adalah dengan kebijakan. Kebijakan adalah dokumen yang menjelaskan prinsip mana yang dapat mengakses sumber daya mana. Kebijakan yang melekat pada identitas IAM disebut kebijakan berbasis identitas (atau kebijakan IAM), dan kebijakan yang melekat pada jenis sumber daya lainnya disebut kebijakan sumber daya. AWS KMSkebijakan sumber daya untuk kunci KMS disebut kebijakan kunci. Semua kunci KMS memiliki kebijakan kunci.

Untuk mengontrol akses ke AWS KMS alias Anda, gunakan kebijakan IAM. Untuk mengizinkan prinsipal membuat alias, Anda harus memberikan izin ke alias dalam kebijakan IAM dan izin ke kunci dalam kebijakan kunci. Untuk detailnya, lihat Mengontrol akses ke alias.

Untuk mengontrol akses ke kunci KMS, Anda dapat menggunakan mekanisme kebijakan berikut.

- Kebijakan kunci — Setiap kunci KMS memiliki kebijakan utama. Ini adalah mekanisme utama untuk mengendalikan akses ke kunci KMS. Anda dapat menggunakan kebijakan kunci sendiri untuk mengontrol akses, yang berarti cakupan penuh akses ke kunci KMS didefinisikan dalam satu dokumen (kebijakan kunci). Untuk informasi selengkapnya tentang cara menggunakan kebijakan kunci, lihat Kebijakan utama.
- Kebijakan IAM — Anda dapat menggunakan kebijakan IAM dalam kombinasi dengan kebijakan utama dan hibah untuk mengontrol akses ke kunci KMS. Mengontrol akses dengan cara ini memungkinkan Anda mengelola semua izin untuk identitas IAM Anda di IAM. Untuk menggunakan kebijakan IAM untuk mengizinkan akses ke kunci KMS, kebijakan kunci harus secara eksplisit mengizinkannya. Untuk informasi selengkapnya tentang cara menggunakan kebijakan IAM, lihat Kebijakan IAM.
- Hibah — Anda dapat menggunakan hibah dalam kombinasi dengan kebijakan utama dan kebijakan IAM untuk mengizinkan akses ke kunci KMS. Mengontrol akses dengan cara ini memungkinkan Anda mengizinkan akses ke kunci KMS dalam kebijakan kunci, dan mengizinkan identitas mendelegasikan akses mereka ke orang lain. Untuk informasi selengkapnya tentang cara menggunakan izin, lihat Hibah di AWS KMS.

Kunci KMS milik AWS akun tempat mereka dibuat. Namun, tidak ada identitas atau prinsipal, termasuk pengguna root AWS akun, yang memiliki izin untuk menggunakan atau mengelola kunci KMS kecuali izin tersebut secara eksplisit diberikan dalam kebijakan utama, kebijakan IAM, atau hibah. Identitas IAM yang membuat kunci KMS tidak dianggap sebagai pemilik kunci dan mereka tidak secara otomatis memiliki izin untuk menggunakan atau mengelola kunci KMS yang mereka buat. Seperti identitas lainnya, pembuat kunci perlu mendapatkan izin melalui kebijakan kunci, kebijakan IAM, atau hibah. Namun, identitas yang memiliki `kms:CreateKey` izin dapat mengatur kebijakan kunci awal dan memberi diri mereka izin untuk menggunakan atau mengelola kunci.

Topik berikut memberikan rincian tentang bagaimana Anda dapat menggunakan AWS Identity and Access Management (IAM) dan AWS KMS izin untuk membantu mengamankan sumber daya Anda dengan mengontrol siapa yang dapat mengaksesnya.

Topik

Konsep dalam kontrol AWS KMS akses

Pelajari konsep yang digunakan dalam diskusi kontrol akses diAWS KMS.

Topik

Autentikasi

Otentikasi adalah proses verifikasi identitas Anda. Untuk mengirim permintaanAWS KMS, Anda harus masuk AWS menggunakan AWS kredensyal Anda.

Otorisasi

Otorisasi memberikan izin untuk mengirim permintaan untuk membuat, mengelola, atau menggunakan AWS KMS sumber daya. Misalnya, Anda harus diberi wewenang untuk menggunakan kunci KMS dalam operasi kriptografi.

Untuk mengontrol akses ke AWS KMS sumber daya Anda, gunakan kebijakan utama, kebijakan IAM, dan hibah. Setiap kunci KMS harus memiliki kebijakan kunci. Jika kebijakan kunci mengizinkannya, Anda juga dapat menggunakan kebijakan dan hibah IAM untuk memberikan akses kepada prinsipal ke kunci KMS. Untuk menyempurnakan otorisasi, Anda dapat menggunakan kunci kondisi yang mengizinkan atau menolak akses hanya jika permintaan atau sumber daya memenuhi ketentuan yang Anda tentukan. Anda juga dapat mengizinkan akses ke kepala sekolah yang Anda percayai pada orang lain. Akun AWS

Mengautentikasi dengan identitas

Autentikasi adalah cara Anda untuk masuk ke AWS menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. Pengguna AWS IAM Identity Center Pengguna (Pusat Identitas IAM), autentikasi Single Sign-On perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal akses AWS. Untuk informasi selengkapnya tentang cara masuk ke AWS, lihat Cara masuk ke Akun AWS dalam Panduan Pengguna AWS Sign-In.

Jika Anda mengakses AWS secara terprogram, AWS memberikan Kit Pengembangan Perangkat Lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan peralatan AWS, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang cara menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan API AWS dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS dalam Panduan Pengguna IAM.

Pengguna root Akun AWS

Ketika membuat Akun AWS, Anda memulai dengan satu identitas masuk yang memiliki akses penuh ke semua Layanan AWS dan sumber daya di akun tersebut. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk menggunakan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas lengkap yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Praktik terbaiknya adalah mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran ini memberikan kredensial sementara.

Untuk pengelolaan akses terpusat, sebaiknya Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apa yang dimaksud Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center.

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM merupakan identitas dalam Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara dalam AWS Management Console dengan berganti peran. Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat Rangkaian izin dalam Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat menyertakan kebijakan secara langsung ke sumber daya (bukan menggunakan peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Saat menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya diajukan saat layanan menerima permintaan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Meneruskan sesi akses.
  - Peran IAM – Peran layanan adalah peran IAM yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus

peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

- Peran terkait layanan – Peran terkait layanan adalah tipe peran layanan yang terkait dengan Layanan AWS. Layanan tersebut dapat mengambil peran untuk melakukan sebuah tindakan atas nama Anda. Peran terkait layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan API AWS CLI atau AWS. Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan peran AWS ke instans EC2 dan menyediakannya bagi semua aplikasinya, Anda dapat membuat profil instans yang dilampirkan ke instans tersebut. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika seorang pengguna utama (pengguna, pengguna root, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat Gambaran Umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses terhadap apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat

membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau API AWS.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola meliputi kebijakan yang dikelola AWS dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat Memilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan AWS KMS kunci adalah kebijakan berbasis sumber daya yang mengontrol akses ke kunci KMS. Setiap kunci KMS harus memiliki kebijakan kunci. Anda dapat menggunakan mekanisme otorisasi lain untuk mengizinkan akses ke kunci KMS, tetapi hanya jika kebijakan kunci mengizinkannya. (Anda dapat menggunakan kebijakan IAM untuk menolak akses ke kunci KMS meskipun kebijakan kunci tidak secara eksplisit mengizinkannya.)

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya, seperti kunci KMS, untuk mengontrol akses ke sumber daya tertentu. Kebijakan berbasis sumber daya mendefinisikan tindakan yang dapat dilakukan oleh prinsipal tertentu pada sumber daya itu dan dalam kondisi apa. Anda tidak menentukan sumber daya dalam kebijakan berbasis sumber daya, tetapi Anda harus menentukan prinsipal, seperti akun, pengguna, peran, pengguna gabungan, atau. Layanan AWS Kebijakan berbasis sumber daya adalah kebijakan inline yang terletak di layanan yang mengelola sumber daya. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM,

seperti kebijakan `AWSKeyManagementServicePowerUser` terkelola, dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

AWS KMStidak mendukung ACL.

Jenis kebijakan lainnya

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa akun Akun AWS yang dimiliki bisnis Anda secara terpusat. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran

dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan ketika ada beberapa jenis kebijakan, lihat Logika evaluasi kebijakan dalam Panduan Pengguna IAM.

Sumber daya AWS KMS

DalamAWS KMS, sumber daya utama adalah AWS KMS key. AWS KMSjuga mendukung alias, sumber daya independen yang menyediakan nama ramah untuk kunci KMS. Beberapa AWS KMS operasi memungkinkan Anda menggunakan alias untuk mengidentifikasi kunci KMS.

Setiap instance kunci atau alias KMS memiliki Amazon Resource Name (ARN) unik dengan format standar. Dalam sumber daya AWS KMS, nama layanan AWS adalah `kms`.

- AWS KMS key

  Format ARN:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`Wilayah AWS`*`:`*`Akun AWS ID`*`:key/`*`key ID`*

  ARN contoh:

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

- Alias

  Format ARN:

  `arn:`*`AWS partition name`*`:`*`AWS service name`*`:`*`Wilayah AWS`*`:`*`Akun AWS ID`*`:alias/`*`alias name`*

  ARN contoh:

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

AWS KMS menyediakan sekumpulan operasi API untuk berfungsi dengan sumber daya AWS KMS. Untuk informasi selengkapnya tentang mengidentifikasi kunci KMS dalam operasi AWS Management Console dan AWS KMS API, lihatPengidentifikasi kunci () KeyId. Untuk daftar operasi AWS KMS, lihat Referensi API AWS Key Management Service.

Kebijakan utama di AWS KMS

Kebijakan kunci adalah kebijakan sumber daya untuk sebuah AWS KMS key. Kebijakan utama adalah cara utama untuk mengontrol akses ke kunci KMS. Setiap kunci KMS harus memiliki satu kebijakan kunci. Pernyataan dalam kebijakan kunci menentukan siapa yang memiliki izin untuk menggunakan kunci KMS dan bagaimana mereka dapat menggunakannya. Anda juga dapat menggunakan kebijakan dan hibah IAM untuk mengontrol akses ke kunci KMS, tetapi setiap kunci KMS harus memiliki kebijakan utama.

Tidak ada AWS prinsipal, termasuk pengguna root akun atau pembuat kunci, yang memiliki izin untuk kunci KMS kecuali mereka secara eksplisit diizinkan, dan tidak pernah ditolak, dalam kebijakan utama, kebijakan IAM, atau hibah.

Kecuali kebijakan kunci secara eksplisit mengizinkannya, Anda tidak dapat menggunakan kebijakan IAM untuk mengizinkan akses ke kunci KMS. Tanpa izin dari kebijakan utama, kebijakan IAM yang mengizinkan izin tidak berpengaruh. (Anda dapat menggunakan kebijakan IAM untuk menolak izin ke kunci KMS tanpa izin dari kebijakan kunci.) Kebijakan kunci default mengaktifkan kebijakan IAM. Untuk mengaktifkan kebijakan IAM dalam kebijakan utama Anda, tambahkan pernyataan kebijakan yang dijelaskan diMengizinkan akses ke Akun AWS dan mengaktifkan kebijakan IAM.

Tidak seperti kebijakan IAM, yang bersifat global, kebijakan kunci bersifat Regional. Kebijakan kunci mengontrol akses hanya ke kunci KMS di Wilayah yang sama. Ini tidak berpengaruh pada kunci KMS di Wilayah lain.

Topik

- Membuat kebijakan utama
- Kebijakan kunci default
- Melihat kebijakan kunci
- Mengubah kebijakan kunci
- Izin untuk AWS layanan dalam kebijakan utama

Membuat kebijakan utama

Anda dapat membuat dan mengelola kebijakan utama di AWS KMS konsol, dengan menggunakan operasi AWS KMS API, seperti CreateKey, ReplicateKey, dan PutKeyPolicy, atau dengan menggunakan AWS CloudFormation templat.

Saat Anda membuat kunci KMS di AWS KMS konsol, konsol akan memandu Anda melalui langkah-langkah membuat kebijakan kunci berdasarkan kebijakan kunci default untuk konsol. Bila Anda menggunakan `CreateKey` atau `ReplicateKey` API, jika Anda tidak menentukan kebijakan kunci, API ini menerapkan kebijakan kunci default untuk kunci yang dibuat secara terprogram. Saat Anda menggunakan `PutKeyPolicy` API, Anda diminta untuk menentukan kebijakan kunci.

Setiap dokumen kebijakan dapat memiliki satu atau lebih pernyataan kebijakan. Contoh berikut menunjukkan dokumen kebijakan kunci yang valid dengan satu pernyataan kebijakan.

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

Topik

- Format kebijakan utama
- Elemen dalam kebijakan utama
- Contoh kebijakan kunci

Format kebijakan utama

Dokumen kebijakan utama harus sesuai dengan aturan berikut:

- Hingga 32 kilobyte (32.768 byte)
- `Sid`Elemen dalam pernyataan kebijakan kunci dapat mencakup spasi. (Spasi dilarang dalam `Sid` elemen dokumen kebijakan IAM.)

Dokumen kebijakan utama hanya dapat menyertakan karakter berikut:

- Karakter ASCII yang dapat dicetak
- Karakter yang dapat dicetak dalam set karakter Suplemen Latin dan Latin-1 Dasar
- Karakter khusus tab (`\u0009`), line feed (`\u000A`), dan carriage return (`\u000D`)

Elemen dalam kebijakan utama

Dokumen kebijakan utama harus memiliki elemen-elemen berikut:

Versi

Menentukan versi dokumen kebijakan utama. Setel versi ke `2012-10-17` (versi terbaru).

Pernyataan

Melampirkan pernyataan kebijakan. Dokumen kebijakan utama harus memiliki setidaknya satu pernyataan.

Setiap pernyataan kebijakan utama terdiri dari hingga enam elemen. Itu`Effect`,`Principal`,`Action`, dan `Resource` elemen diperlukan.

Sid

(Opsional) Pernyataan identifier (`Sid`) string arbitrer yang dapat Anda gunakan untuk menggambarkan pernyataan. Kebijakan `Sid` dalam kunci dapat mencakup spasi. (Anda tidak dapat menyertakan spasi dalam `Sid` elemen kebijakan IAM.)

Efek

(Wajib) Menentukan apakah akan mengizinkan atau menolak izin dalam pernyataan kebijakan. Nilai-nilai yang valid adalah `Allow` atau `Deny`. Jika Anda tidak secara eksplisit mengizinkan

akses ke kunci KMS, akses secara implisit ditolak. Anda juga dapat secara eksplisit menolak akses ke kunci KMS. Anda dapat melakukan langkah ini untuk memastikan bahwa pengguna tidak dapat mengaksesnya, meski kebijakan yang berbeda mengizinkan aksesnya.

Utama

(Wajib) Prinsipal adalah identitas yang mendapatkan izin yang ditentukan dalam pernyataan kebijakan. Anda dapat menentukan Akun AWS, pengguna IAM, peran IAM, dan beberapa AWS layanan sebagai prinsipal dalam kebijakan utama. Grup pengguna IAM bukan prinsipal yang valid dalam jenis kebijakan apa pun.

Nilai tanda bintang, seperti `"AWS": "*"` mewakili semua AWS identitas di semua akun.

> **Important**
>
> Jangan menyetel Principal ke tanda bintang (*) dalam pernyataan kebijakan kunci apa pun yang mengizinkan izin kecuali Anda menggunakan kondisi untuk membatasi kebijakan utama. Tanda bintang memberikan setiap identitas di setiap Akun AWS izin untuk menggunakan kunci KMS, kecuali pernyataan kebijakan lain secara eksplisit menyangkalnya. Pengguna lain Akun AWS dapat menggunakan kunci KMS Anda setiap kali mereka memiliki izin yang sesuai di akun mereka sendiri.

> **Note**
>
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Ketika prinsipal dalam pernyataan kebijakan kunci adalah Akun AWS prinsipal yang dinyatakan sebagai`arn:aws:iam::111122223333:root"`, pernyataan kebijakan tidak memberikan izin kepada kepala sekolah IAM mana pun. Sebagai gantinya, ini memberikan Akun AWS izin untuk menggunakan kebijakan IAM untuk mendelegasikan izin yang ditentukan dalam kebijakan kunci. (Prinsipal dalam `arn:aws:iam::111122223333:root"` format tidak mewakili pengguna root AWS akun, meskipun menggunakan “root” dalam pengenal akun. Namun, prinsipal akun mewakili akun dan administratornya, termasuk pengguna root akun.)

Ketika prinsipal adalah yang lain Akun AWS atau prinsipalnya, izin hanya berlaku jika akun diaktifkan di Wilayah dengan kunci KMS dan kebijakan kunci. Untuk informasi tentang Wilayah yang tidak diaktifkan secara default ("opt-in Regions"), lihat Mengelola Wilayah AWS di Referensi Umum AWS.

Untuk mengizinkan yang berbeda Akun AWS atau prinsipal menggunakan kunci KMS, Anda harus memberikan izin dalam kebijakan kunci dan dalam kebijakan IAM di akun lain. Lihat perinciannya di Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

Tindakan

(Wajib) Tentukan operasi API untuk mengizinkan atau menolak. Misalnya, `kms:Encrypt` tindakan tersebut sesuai dengan operasi AWS KMS Enkripsi. Anda dapat mencantumkan beberapa tindakan dalam pernyataan kebijakan. Untuk informasi selengkapnya, lihat Referensi izin.

Sumber Daya

(Wajib) Dalam kebijakan kunci, nilai elemen Sumber Daya adalah`"*"`, yang berarti "kunci KMS ini." Tanda bintang (`"*"`) mengidentifikasi kunci KMS tempat kebijakan kunci dilampirkan.

> **Note**
>
> Jika `Resource` elemen yang diperlukan hilang dari pernyataan kebijakan kunci, pernyataan kebijakan tidak berpengaruh. Pernyataan kebijakan kunci tanpa `Resource` elemen tidak berlaku untuk kunci KMS apa pun.
> Ketika pernyataan kebijakan kunci kehilangan `Resource` elemennya, AWS KMS konsol melaporkan kesalahan dengan benar, tetapi PutKeyPolicyAPI CreateKeydan berhasil, meskipun pernyataan kebijakan tidak efektif.

Ketentuan

(Opsional) Ketentuan menentukan persyaratan yang harus dipenuhi agar kebijakan utama berlaku. Dengan kondisi, AWS dapat mengevaluasi konteks permintaan API untuk menentukan apakah pernyataan kebijakan berlaku atau tidak.

Untuk menentukan kondisi, Anda menggunakan kunci kondisi yang telah ditentukan. AWS KMS mendukung kunci kondisi AWS global dan kunci AWS KMS kondisi. Untuk mendukung

kontrol akses berbasis atribut (ABAC), AWS KMS berikan kunci kondisi yang mengontrol akses ke kunci KMS berdasarkan tag dan alias. Lihat perinciannya di ABAC untuk AWS KMS.

Format untuk suatu kondisi adalah:

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

seperti:

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

Untuk informasi selengkapnya tentang sintaks AWS kebijakan, lihat Referensi Kebijakan AWS IAM di Panduan Pengguna IAM.

Contoh kebijakan kunci

Contoh berikut menunjukkan kebijakan kunci lengkap untuk kunci KMS enkripsi simetris. Anda dapat menggunakannya untuk referensi saat Anda membaca tentang konsep kebijakan utama dalam Bab ini. Kebijakan kunci ini menggabungkan contoh pernyataan kebijakan dari bagian kebijakan kunci default sebelumnya ke kebijakan kunci tunggal yang menyelesaikan berikut ini:

- Memungkinkan contoh Akun AWS, 111122223333, akses penuh ke kunci KMS. Ini memungkinkan akun dan administratornya, termasuk pengguna root akun (untuk keadaan darurat), untuk menggunakan kebijakan IAM di akun untuk memungkinkan akses ke kunci KMS.
- Memungkinkan peran `ExampleAdminRole` IAM untuk mengelola kunci KMS.
- Memungkinkan peran `ExampleUserRole` IAM untuk menggunakan kunci KMS.

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

Kebijakan kunci default

Saat membuat kunci KMS, Anda dapat menentukan kebijakan kunci untuk kunci KMS baru. Jika Anda tidak menyediakannya, AWS KMS buatlah satu untuk Anda. Kebijakan kunci default yang AWS KMS menggunakan berbeda tergantung pada apakah Anda membuat kunci di AWS KMS konsol atau Anda menggunakan AWS KMS API.

Kebijakan kunci default saat Anda membuat kunci KMS secara terprogram

Saat Anda membuat kunci KMS secara terprogram dengan AWS KMS API (termasuk dengan menggunakan AWS SDK, AWS Command Line Interfaceatau AWS Tools for PowerShell), dan Anda tidak menentukan kebijakan kunci, AWS KMS menerapkan kebijakan kunci default yang sangat sederhana. Kebijakan kunci default ini memiliki satu pernyataan kebijakan yang memberikan izin Akun AWS yang memiliki kunci KMS untuk menggunakan kebijakan IAM untuk mengizinkan akses ke semua AWS KMS operasi pada kunci KMS. Untuk informasi selengkapnya tentang pernyataan kebijakan ini, lihat Mengizinkan akses ke Akun AWS dan mengaktifkan kebijakan IAM.

Kebijakan kunci default saat Anda membuat kunci KMS dengan AWS Management Console

Saat Anda membuat kunci KMS dengan AWS Management Console, kebijakan kunci dimulai dengan pernyataan kebijakan yang memungkinkan akses ke Akun AWS dan mengaktifkan kebijakan IAM. Konsol kemudian menambahkan pernyataan administrator kunci, pernyataanpengguna kunci, dan (untuk sebagian besar jenis kunci) pernyataan yang memungkinkan prinsipal menggunakan kunci

KMS dengan layanan lain. AWS Anda dapat menggunakan fitur AWS KMS konsol untuk menentukan pengguna IAM, iamRoles, dan Akun AWS siapa administrator kunci dan mereka yang merupakan pengguna kunci (atau keduanya).

Izin

- Mengizinkan akses ke Akun AWS dan mengaktifkan kebijakan IAM
- Memungkinkan administrator kunci untuk mengelola kunci KMS
- Memungkinkan pengguna kunci untuk menggunakan kunci KMS
  - Memungkinkan pengguna kunci untuk menggunakan kunci KMS untuk operasi kriptografi
  - Memungkinkan pengguna kunci untuk menggunakan kunci KMS dengan layanan AWS

Mengizinkan akses ke Akun AWS dan mengaktifkan kebijakan IAM

Pernyataan kebijakan kunci default berikut sangat penting.

- Ini memberikan Akun AWS yang memiliki kunci KMS akses penuh ke kunci KMS.

  Tidak seperti kebijakan AWS sumber daya lainnya, kebijakan AWS KMS kunci tidak secara otomatis memberikan izin ke akun atau identitasnya. Untuk memberikan izin kepada administrator akun, kebijakan utama harus menyertakan pernyataan eksplisit yang memberikan izin ini, seperti ini.
- Ini memungkinkan akun untuk menggunakan kebijakan IAM untuk memungkinkan akses ke kunci KMS, selain kebijakan utama.

  Tanpa izin ini, kebijakan IAM yang memungkinkan akses ke kunci tidak efektif, meskipun kebijakan IAM yang menolak akses ke kunci masih efektif.
- Ini mengurangi risiko kunci menjadi tidak terkendali dengan memberikan izin kontrol akses ke administrator akun, termasuk pengguna root akun, yang tidak dapat dihapus.

Pernyataan kebijakan kunci berikut adalah seluruh kebijakan kunci default untuk kunci KMS yang dibuat secara terprogram. Ini adalah pernyataan kebijakan pertama dalam kebijakan kunci default untuk kunci KMS yang dibuat di AWS KMS konsol.

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
   },
  "Action": "kms:*",
  "Resource": "*"
}
```

Mengizinkan kebijakan IAM untuk mengizinkan akses ke kunci KMS.

Pernyataan kebijakan utama yang ditunjukkan di atas memberikan izin Akun AWS yang memiliki kunci untuk menggunakan kebijakan IAM, serta kebijakan utama, untuk mengizinkan semua tindakan (`kms:*`) pada kunci KMS.

Prinsip dalam pernyataan kebijakan utama ini adalah pokok akun, yang diwakili oleh ARN dalam format ini:. `arn:aws:iam::account-id:root` Prinsipal akun mewakili AWS akun dan administratornya.

Ketika prinsipal dalam pernyataan kebijakan utama adalah pokok akun, pernyataan kebijakan tidak memberikan izin utama IAM untuk menggunakan kunci KMS. Sebagai gantinya, akun dapat menggunakan kebijakan IAM untuk mendelegasikan izin yang ditentukan dalam pernyataan kebijakan. Pernyataan kebijakan kunci default ini memungkinkan akun menggunakan kebijakan IAM untuk mendelegasikan izin untuk semua tindakan (`kms:*`) pada kunci KMS.

Mengurangi risiko kunci KMS menjadi tidak terkendali.

Tidak seperti kebijakan AWS sumber daya lainnya, kebijakan AWS KMS kunci tidak secara otomatis memberikan izin ke akun atau prinsipalnya. Untuk memberikan izin kepada prinsipal apa pun, termasuk prinsipal akun, Anda harus menggunakan pernyataan kebijakan utama yang memberikan izin secara eksplisit. Anda tidak diharuskan untuk memberikan prinsipal akun, atau prinsipal apa pun, akses ke kunci KMS. Namun, memberikan akses ke prinsipal akun membantu Anda mencegah kunci menjadi tidak dapat dikelola.

Misalnya, Anda membuat kebijakan kunci yang hanya memberikan satu pengguna akses ke kunci KMS. Jika Anda kemudian menghapus pengguna itu, kunci menjadi tidak dapat dikelola dan Anda harus menghubungi AWS Support untuk mendapatkan kembali akses ke kunci KMS.

Pernyataan kebijakan kunci yang ditunjukkan di atas memberikan izin untuk mengontrol kunci ke prinsipal akun, yang mewakili Akun AWS dan administratornya, termasuk pengguna root akun. Pengguna root akun adalah satu-satunya prinsipal yang tidak dapat dihapus kecuali Anda menghapus Akun AWS. Praktik terbaik IAM mencegah bertindak atas nama pengguna root akun,

kecuali dalam keadaan darurat. Namun, Anda mungkin perlu bertindak sebagai pengguna root akun jika Anda menghapus semua pengguna dan peran lain dengan akses ke kunci KMS.

Memungkinkan administrator kunci untuk mengelola kunci KMS

Kebijakan kunci default yang dibuat oleh konsol tersebut mengizinkan Anda memilih pengguna IAM dan peran dalam akun dan membuatnya administrator kunci. Pernyataan ini disebut pernyataan administrator kunci. Administrator kunci memiliki izin untuk mengelola kunci KMS, tetapi tidak memiliki izin untuk menggunakan kunci KMS dalam operasi kriptografi. Anda dapat menambahkan pengguna dan peran IAM ke daftar administrator kunci saat Anda membuat kunci KMS dalam tampilan default atau tampilan kebijakan.

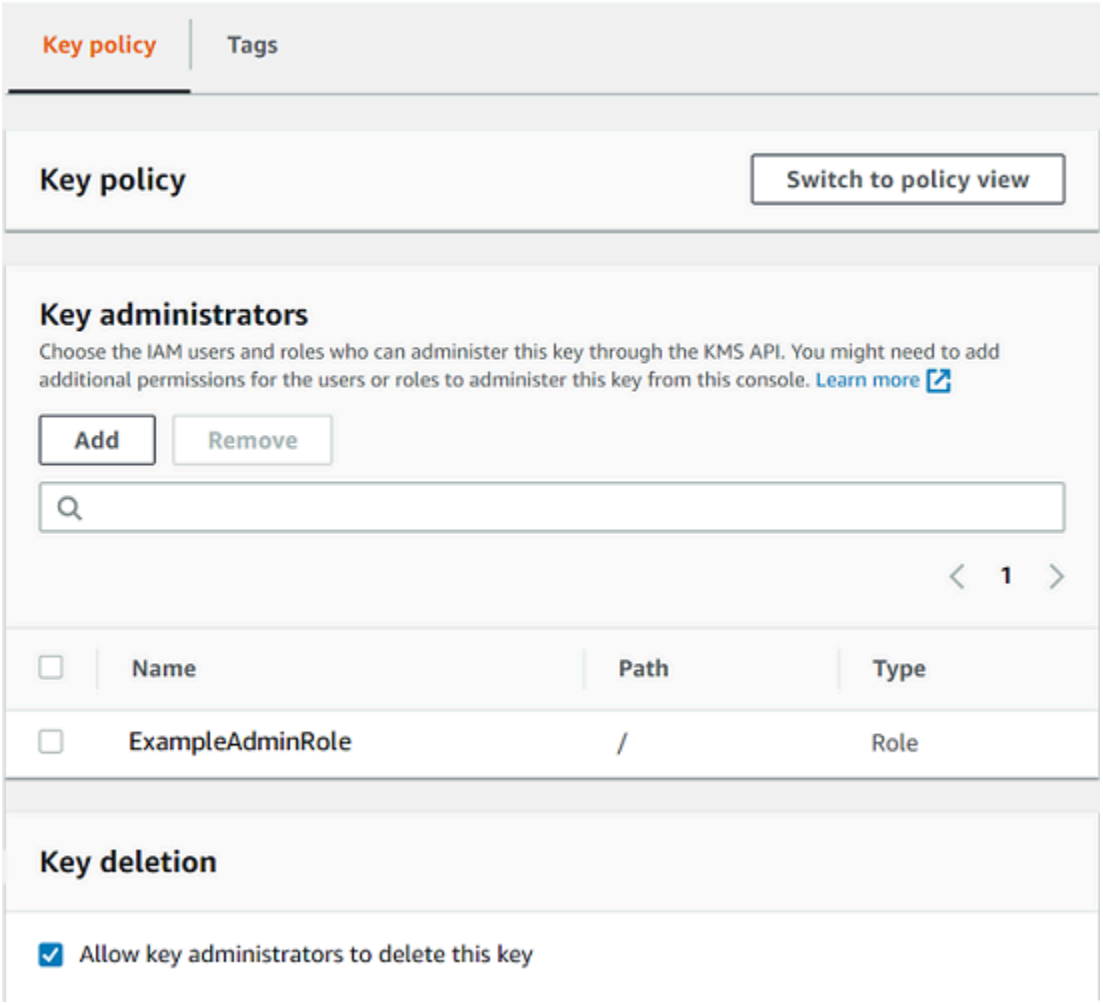
Warning

Karena administrator kunci memiliki izin untuk mengubah kebijakan kunci dan membuat hibah, mereka dapat memberikan AWS KMS izin kepada diri mereka sendiri dan orang lain yang tidak ditentukan dalam kebijakan ini.
Prinsipal yang memiliki izin untuk mengelola tag dan alias juga dapat mengontrol akses ke kunci KMS. Lihat perinciannya di ABAC untuk AWS KMS.

Note

Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Contoh berikut menunjukkan pernyataan administrator kunci dalam tampilan default AWS KMS konsol.



Berikut ini adalah contoh pernyataan administrator kunci dalam tampilan kebijakan AWS KMS konsol. Pernyataan administrator kunci ini adalah untuk kunci KMS enkripsi simetris wilayah tunggal.

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

Pernyataan administrator kunci default untuk kunci KMS yang paling umum, kunci KMS enkripsi simetris wilayah tunggal, memungkinkan izin berikut. Untuk informasi rinci tentang setiap izin, lihatAWS KMS izin.

Saat Anda menggunakan AWS KMS konsol untuk membuat kunci KMS, konsol menambahkan pengguna dan peran yang Anda tentukan ke `Principal` elemen dalam pernyataan administrator kunci.

Banyak dari izin ini berisi karakter wildcard (*), yang memungkinkan semua izin yang dimulai dengan kata kerja yang ditentukan. Akibatnya, ketika AWS KMS menambahkan operasi API baru, administrator kunci secara otomatis diizinkan untuk menggunakannya. Anda tidak perlu memperbarui kebijakan utama Anda untuk menyertakan operasi baru. Jika Anda lebih suka membatasi administrator kunci ke set operasi API tetap, Anda dapat mengubah kebijakan kunci Anda.

`kms:Create*`

Memungkinkan `kms:CreateAlias`dan `kms:CreateGrant`. (`kms:CreateKey`Izin hanya berlaku dalam kebijakan IAM.)

`kms:Describe*`

Memungkinkan `kms:DescribeKey`. `kms:DescribeKey`Izin diperlukan untuk melihat halaman detail kunci untuk kunci KMS di. AWS Management Console

`kms:Enable*`

Memungkinkan `kms:EnableKey`. Untuk kunci KMS enkripsi simetris, ini juga memungkinkan. `kms:EnableKeyRotation`

`kms:List*`

Memungkinkan `kms:ListGrants`, `kms:ListKeyPolicies`, dan `kms:ListResourceTags`. (`kms:ListKeys`Izin `kms:ListAliases` dan, yang diperlukan untuk melihat kunci KMS di AWS Management Console, hanya berlaku dalam kebijakan IAM.)

`kms:Put*`

Memungkinkan `kms:PutKeyPolicy`. Izin ini memungkinkan administrator kunci untuk mengubah kebijakan kunci untuk kunci KMS ini.

`kms:Update*`

Memungkinkan `kms:UpdateAlias`dan `kms:UpdateKeyDescription`. Untuk kunci Multi-region, ini memungkinkan `kms:UpdatePrimaryRegion`pada kunci KMS ini.

`kms:Revoke*`

Memungkinkan `kms:RevokeGrant`, yang memungkinkan administrator kunci untuk menghapus hibah meskipun mereka bukan kepala sekolah pensiun dalam hibah.

`kms:Disable*`

Memungkinkan `kms:DisableKey`. Untuk kunci KMS enkripsi simetris, ini juga memungkinkan. `kms:DisableKeyRotation`

`kms:Get*`

Memungkinkan `kms:GetKeyPolicy`dan `kms:GetKeyRotationStatus`. Untuk kunci KMS dengan bahan kunci impor, memungkinkan `kms:GetParametersForImport`. Untuk kunci KMS asimetris, ini memungkinkan. `kms:GetPublicKey` `kms:GetKeyPolicy`Izin diperlukan untuk melihat kebijakan kunci kunci KMS di. AWS Management Console

`kms:Delete*`

Memungkinkan `kms:DeleteAlias`. Untuk kunci dengan bahan kunci impor, memungkinkan `kms:DeleteImportedKeyMaterial`. `kms:Delete*`Izin tidak mengizinkan administrator kunci untuk menghapus kunci KMS ()`ScheduleKeyDeletion`.

`kms:TagResource`

Memungkinkan `kms:TagResource`, yang memungkinkan administrator kunci untuk menambahkan tag ke kunci KMS. Karena tag juga dapat digunakan untuk mengontrol akses ke kunci KMS, izin ini dapat memungkinkan administrator untuk mengizinkan atau menolak akses ke kunci KMS. Lihat perinciannya di ABAC untuk AWS KMS.

`kms:UntagResource`

Memungkinkan `kms:UntagResource`, yang memungkinkan administrator kunci untuk menghapus tag dari kunci KMS. Karena tag dapat digunakan untuk mengontrol akses ke kunci, izin ini dapat memungkinkan administrator untuk mengizinkan atau menolak akses ke kunci KMS. Lihat perinciannya di ABAC untuk AWS KMS.

`kms:ScheduleKeyDeletion`

Memungkinkan `kms:ScheduleKeyDeletion`, yang memungkinkan administrator kunci untuk menghapus kunci KMS ini. Untuk menghapus izin ini, kosongkan opsi Izinkan administrator kunci untuk menghapus opsi kunci ini.

`kms:CancelKeyDeletion`

Memungkinkan `kms:CancelKeyDeletion`, yang memungkinkan administrator kunci untuk membatalkan penghapusan kunci KMS ini. Untuk menghapus izin ini, kosongkan opsi Izinkan administrator kunci untuk menghapus opsi kunci ini.

AWS KMS menambahkan izin berikut ke pernyataan administrator kunci default saat Anda membuat kunci tujuan khusus.

`kms:ImportKeyMaterial`

`kms:ImportKeyMaterial`Izin memungkinkan administrator kunci untuk mengimpor materi kunci ke dalam kunci KMS. Izin ini disertakan dalam kebijakan kunci hanya jika Anda membuat kunci KMS tanpa materi kunci.

`kms:ReplicateKey`

`kms:ReplicateKey`Izin ini memungkinkan administrator kunci untuk membuat replika kunci utama Multi-wilayah di Wilayah yang berbeda. AWS Izin ini disertakan dalam kebijakan kunci hanya jika Anda membuat kunci primer atau replika Multi-wilayah.

`kms:UpdatePrimaryRegion`

`kms:UpdatePrimaryRegion`Izin ini memungkinkan administrator kunci untuk mengubah kunci replika Multi-region menjadi kunci utama Multi-region. Izin ini disertakan dalam kebijakan kunci hanya jika Anda membuat kunci primer atau replika Multi-wilayah.

Memungkinkan pengguna kunci untuk menggunakan kunci KMS

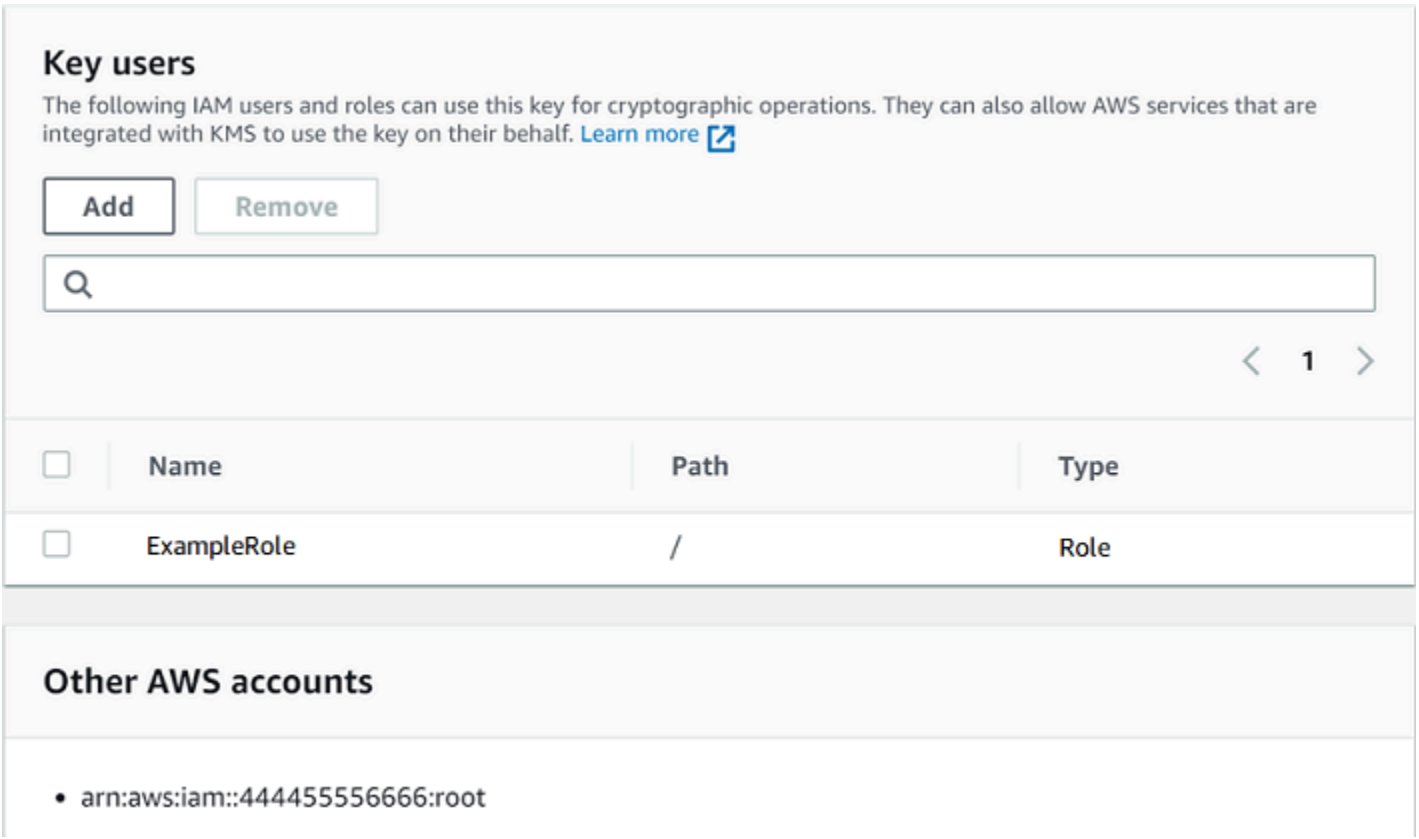
Kebijakan kunci default yang dibuat konsol untuk kunci KMS memungkinkan Anda memilih pengguna IAM dan peran IAM di akun, dan eksternal Akun AWS, dan menjadikannya pengguna utama.

Konsol tersebut menambahkan dua pernyataan kebijakan untuk kebijakan kunci bagi pengguna kunci.

- Gunakan kunci KMS secara langsung — Pernyataan kebijakan kunci pertama memberi pengguna kunci izin untuk menggunakan kunci KMS secara langsung untuk semua operasi kriptografi yang didukung untuk jenis kunci KMS tersebut.
- Gunakan kunci KMS dengan AWS layanan — Pernyataan kebijakan kedua memberikan izin kepada pengguna kunci untuk mengizinkan AWS layanan yang terintegrasi dengan menggunakan kunci KMS atas nama mereka AWS KMS untuk melindungi sumber daya, seperti bucket Amazon S3 dan tabel Amazon DynamoDB.

Anda dapat menambahkan pengguna IAM, peran IAM, dan lainnya Akun AWS ke daftar pengguna utama saat Anda membuat kunci KMS. Anda juga dapat mengedit daftar dengan tampilan default konsol tersebut untuk kebijakan kunci, seperti yang ditunjukkan pada gambar berikut. Tampilan default untuk kebijakan kunci pada halaman detail kunci. Untuk informasi selengkapnya tentang mengizinkan pengguna lain Akun AWS untuk menggunakan kunci KMS, lihatMemungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

> **Note**
>
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.



Pernyataan pengguna kunci default untuk simetris Single-region memungkinkan izin berikut. Untuk informasi rinci tentang setiap izin, lihatAWS KMS izin.

Saat Anda menggunakan AWS KMS konsol untuk membuat kunci KMS, konsol menambahkan pengguna dan peran yang Anda tentukan ke `Principal` elemen di setiap pernyataan pengguna kunci.

```
{
  "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}}
}
```

Memungkinkan pengguna kunci untuk menggunakan kunci KMS untuk operasi kriptografi

Pengguna kunci memiliki izin untuk menggunakan kunci KMS secara langsung di semua operasi kriptografi yang didukung pada kunci KMS. Mereka juga dapat menggunakan DescribeKeyoperasi untuk mendapatkan informasi rinci tentang kunci KMS di AWS KMS konsol atau dengan menggunakan operasi AWS KMS API.

Secara default, AWS KMS konsol menambahkan pernyataan pengguna kunci seperti yang ada dalam contoh berikut ke kebijakan kunci default. Karena mereka mendukung operasi API yang berbeda, tindakan dalam pernyataan kebijakan untuk kunci KMS enkripsi simetris, kunci KMS HMAC, kunci KMS asimetris untuk enkripsi kunci publik, dan kunci KMS asimetris untuk penandatanganan dan verifikasi sedikit berbeda.

Kunci KMS enkripsi simetris

Konsol menambahkan pernyataan berikut ke kebijakan kunci untuk kunci KMS enkripsi simetris.

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

Kunci HMAC KMS

Konsol menambahkan pernyataan berikut ke kebijakan kunci untuk kunci HMAC KMS.

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

Kunci KMS asimetris untuk enkripsi kunci publik

Konsol menambahkan pernyataan berikut ke kebijakan kunci untuk kunci KMS asimetris dengan penggunaan kunci Enkripsi dan dekripsi.

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

```
}
```

Kunci KMS asimetris untuk penandatanganan dan verifikasi

Konsol menambahkan pernyataan berikut ke kebijakan kunci untuk kunci KMS asimetris dengan penggunaan kunci Tanda dan verifikasi.

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

Tindakan dalam pernyataan ini memberi pengguna kunci izin berikut.

`kms:Encrypt`

Memungkinkan pengguna kunci untuk mengenkripsi data dengan kunci KMS ini.

`kms:Decrypt`

Memungkinkan pengguna kunci untuk mendekripsi data dengan kunci KMS ini.

`kms:DescribeKey`

Memungkinkan pengguna kunci untuk mendapatkan informasi rinci tentang kunci KMS ini termasuk pengenal, tanggal pembuatan, dan status kunci. Ini juga memungkinkan pengguna kunci untuk menampilkan detail tentang kunci KMS di AWS KMS konsol.

`kms:GenerateDataKey*`

Memungkinkan pengguna kunci untuk meminta kunci data simetris atau symmetric data key pair untuk operasi kriptografi sisi klien. Konsol menggunakan karakter wildcard * untuk mewakili izin operasi API berikut: GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair, dan GenerateDataKeyPairWithoutPlaintext. Izin ini hanya berlaku pada kunci KMS simetris yang mengenkripsi kunci data.

km: GenerateMac

Memungkinkan pengguna kunci untuk menggunakan kunci HMAC KMS untuk menghasilkan tag HMAC.

km: GetPublicKey

Memungkinkan pengguna kunci untuk mengunduh kunci publik dari kunci KMS asimetris. Pihak dengan siapa Anda berbagi kunci publik ini dapat mengenkripsi data di luar. AWS KMS Namun, ciphertext tersebut dapat didekripsi hanya dengan memanggil operasi Dekripsi di AWS KMS.

km: * ReEncrypt

Memungkinkan pengguna kunci untuk mengenkripsi ulang data yang awalnya dienkripsi dengan kunci KMS ini, atau menggunakan kunci KMS ini untuk mengenkripsi ulang data yang sebelumnya dienkripsi. ReEncryptOperasi ini membutuhkan akses ke kunci KMS sumber dan tujuan. Untuk mencapai ini, Anda dapat mengizinkan `kms:ReEncryptFrom` izin pada kunci KMS sumber dan `kms:ReEncryptTo` izin pada kunci KMS tujuan. Namun, untuk kesederhanaan, konsol memungkinkan `kms:ReEncrypt*` (dengan karakter `*` wildcard) pada kedua tombol KMS.

KMS: Tanda

Memungkinkan pengguna kunci untuk menandatangani pesan dengan kunci KMS ini.

KMS: Verifikasi

Memungkinkan pengguna kunci untuk memverifikasi tanda tangan dengan kunci KMS ini.

km: VerifyMac

Memungkinkan pengguna kunci untuk menggunakan kunci HMAC KMS untuk memverifikasi tag HMAC.

Memungkinkan pengguna kunci untuk menggunakan kunci KMS dengan layanan AWS

Kebijakan kunci default di konsol juga memberi pengguna kunci izin hibah yang mereka perlukan untuk melindungi data mereka di AWS layanan yang menggunakan hibah. AWS Layanan sering menggunakan hibah untuk mendapatkan izin khusus dan terbatas untuk menggunakan kunci KMS.

Pernyataan kebijakan kunci ini memungkinkan pengguna kunci untuk membuat, melihat, dan mencabut hibah pada kunci KMS, tetapi hanya jika permintaan operasi hibah berasal dari layanan yang AWS terintegrasi dengannya. AWS KMS Kondisi GrantIsFor AWSResource kebijakan

kms: tidak memungkinkan pengguna untuk memanggil operasi hibah ini secara langsung. Ketika pengguna kunci mengizinkannya, AWS layanan dapat membuat hibah atas nama pengguna yang memungkinkan layanan menggunakan kunci KMS untuk melindungi data pengguna.

Pengguna kunci memerlukan izin hibah ini untuk menggunakan kunci KMS mereka dengan layanan terintegrasi, tetapi izin ini tidak cukup. Pengguna kunci juga memerlukan izin untuk menggunakan layanan terintegrasi. Untuk detail tentang memberi pengguna akses ke AWS layanan yang terintegrasi AWS KMS, lihat dokumentasi untuk layanan terintegrasi.

```
{
  "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}}
}
```

Misalnya, pengguna kunci dapat menggunakan izin ini pada kunci KMS dengan cara berikut.

- Gunakan kunci KMS ini dengan Amazon Elastic Block Store (Amazon EBS) Block Store (Amazon EBS) dan Amazon Elastic Compute Cloud (Amazon EC2) untuk melampirkan volume EBS terenkripsi ke instans EC2. Pengguna kunci secara implisit memberikan izin Amazon EC2 untuk menggunakan kunci KMS untuk melampirkan volume terenkripsi ke instance. Untuk informasi selengkapnya, lihat Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS.
- Gunakan kunci KMS ini dengan Amazon Redshift untuk meluncurkan cluster terenkripsi. Pengguna kunci secara implisit memberikan izin Amazon Redshift untuk menggunakan kunci KMS untuk meluncurkan cluster terenkripsi dan membuat snapshot terenkripsi. Untuk informasi selengkapnya, lihat Bagaimana Amazon Redshift menggunakan AWS KMS.
- Gunakan kunci KMS ini dengan AWS layanan lain AWS KMS yang terintegrasi dengan hibah penggunaan untuk membuat, mengelola, atau menggunakan sumber daya terenkripsi dengan layanan tersebut.

Kebijakan kunci default memungkinkan pengguna kunci untuk mendelegasikan izin hibah mereka ke semua layanan terintegrasi yang menggunakan hibah. Namun, Anda dapat membuat kebijakan kunci

khusus yang membatasi izin ke AWS layanan tertentu. Untuk informasi selengkapnya, lihat kunci syarat km: ViaService.

Melihat kebijakan kunci

Anda dapat melihat kebijakan kunci untuk kunci yang dikelola AWS KMS pelanggan atau Kunci yang dikelola AWSdi akun Anda dengan menggunakan AWS Management Console atau GetKeyPolicyoperasi di AWS KMS API. Anda tidak dapat menggunakan teknik ini untuk melihat kebijakan kunci kunci KMS secara berbedaAkun AWS.

Untuk mempelajari selengkapnya tentang kebijakan kunci AWS KMS, lihat Kebijakan utama di AWS KMS. Untuk mempelajari cara menentukan pengguna dan peran mana yang memiliki akses ke kunci KMS, lihatthe section called “Menentukan akses”.

Topik

- Melihat kebijakan kunci (konsol)
- Melihat kebijakan kunci (API AWS KMS)

Melihat kebijakan kunci (konsol)

Pengguna yang berwenang dapat melihat kebijakan kunci untuk kunci terkelola Kunci yang dikelola AWSatau pelanggan pada tab Kebijakan kunci padaAWS Management Console.

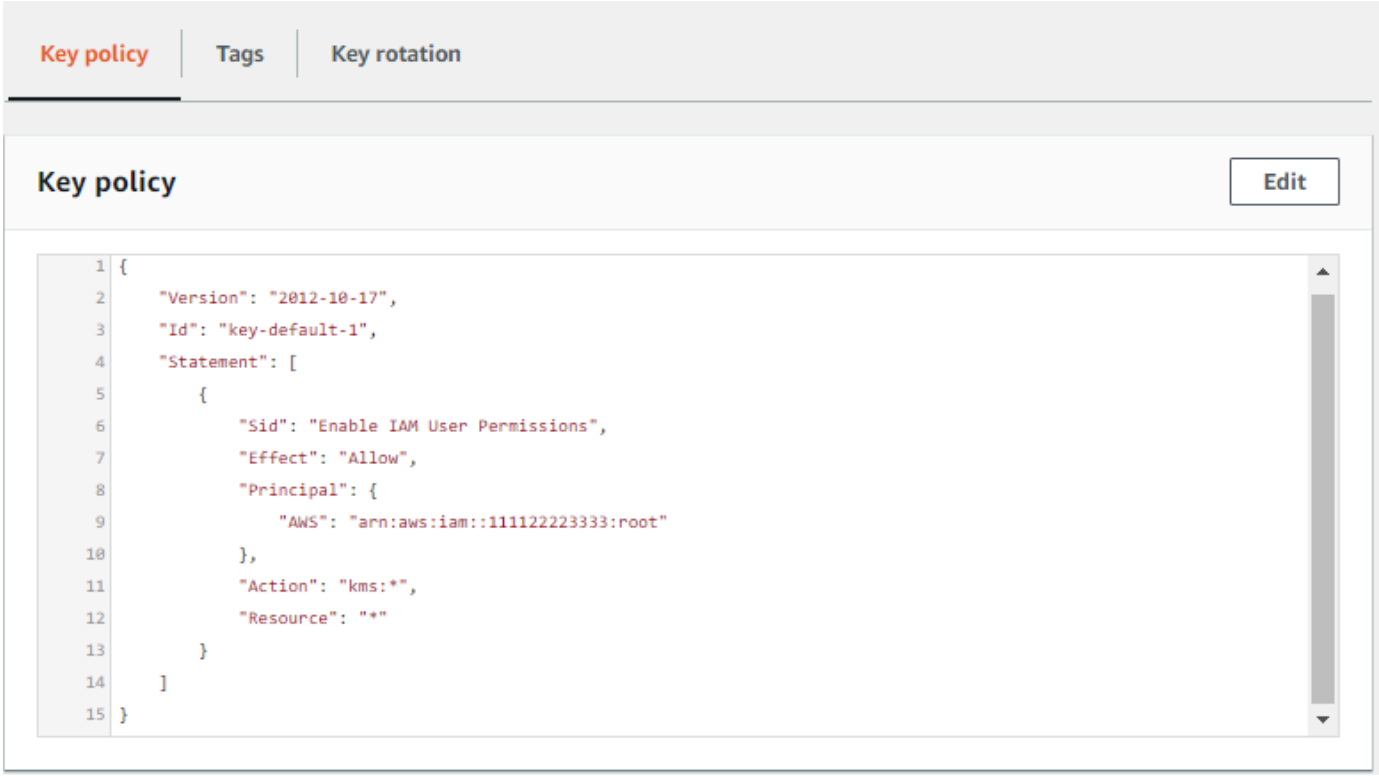
Untuk melihat kebijakan kunci untuk kunci KMS diAWS Management Console, Anda harus memiliki izin kms:, kms: ListAliasesDescribeKey, dan kms:. GetKeyPolicy

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat kunci di akun yang dibuat dan dikelola AWS untuk Anda, pilih Kunci yang dikelola AWS di panel navigasi. Untuk melihat kunci di akun yang Anda buat dan kelola, pilih Kunci yang dikelola pelanggan di panel navigasi.
4. Dalam daftar kunci KMS, pilih alias atau ID kunci dari kunci KMS yang ingin Anda periksa.
5. Pilih tab Kebijakan kunci.

   Pada tab Kebijakan kunci, Anda mungkin melihat dokumen kebijakan kunci. Ini adalah tampilan kebijakan. Dalam pernyataan kebijakan utama, Anda dapat melihat kepala sekolah yang telah

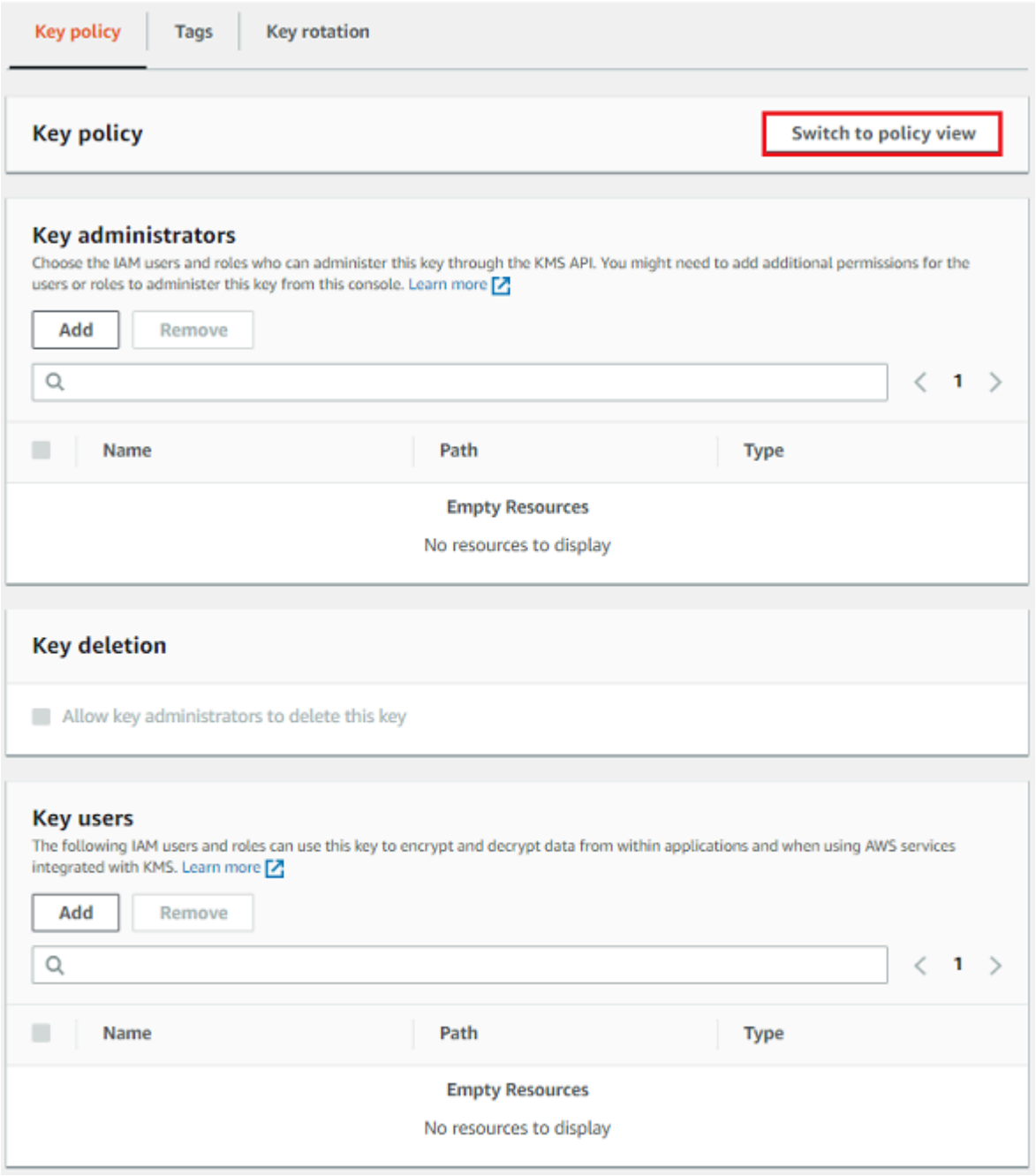
diberi akses ke kunci KMS oleh kebijakan utama, dan Anda dapat melihat tindakan yang dapat mereka lakukan.

Contoh berikut menunjukkan tampilan kebijakan untuk kebijakan kunci default.



Atau, jika Anda membuat kunci KMS diAWS Management Console, Anda akan melihat tampilan default dengan bagian untuk administrator Kunci, Penghapusan kunci, dan Pengguna Kunci. Untuk melihat dokumen kebijakan kunci, pilih Beralih ke tampilan kebijakan.

Contoh berikut menunjukkan tampilan default untuk kebijakan kunci default.



Melihat kebijakan kunci (API AWS KMS)

Untuk mendapatkan kebijakan kunci untuk kunci KMS di AndaAkun AWS, gunakan GetKeyPolicyoperasi di AWS KMS API. Anda tidak dapat menggunakan operasi ini untuk melihat kebijakan kunci di akun yang berbeda.

Contoh berikut menggunakan get-key-policyperintah di AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan AWS SDK apa pun untuk membuat permintaan ini.

Perlu diingat bahwa parameter `PolicyName` diperlukan meskipun `default` adalah satu-satunya nilai yang valid. Selain itu, perintah ini meminta output dalam teks, bukan JSON, agar lebih mudah dilihat.

Sebelum menjalankan perintah ini, ganti contoh ID kunci dengan ID yang valid dari akun Anda.

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

Respons harus serupa dengan berikut ini, yang mengembalikan kebijakan kunci default.

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

Mengubah kebijakan kunci

Anda dapat mengubah kebijakan kunci untuk kunci KMS di Anda Akun AWS dengan menggunakan AWS Management Console atau PutKeyPolicyoperasi. Anda tidak dapat menggunakan teknik ini untuk mengubah kebijakan kunci kunci KMS secara berbedaAkun AWS.

Saat mengubah kebijakan kunci, perhatikan aturan berikut:

- Anda dapat melihat kebijakan kunci untuk kunci yang dikelola pelanggan, tetapi Anda hanya dapat mengubah kebijakan kunci untuk kunci yang dikelola pelanggan. Kunci yang dikelola AWS Kebijakan Kunci yang dikelola AWS dibuat dan dikelola oleh AWS layanan yang membuat kunci KMS di akun Anda. Anda tidak dapat melihat atau mengubah kebijakan kunci untuk file Kunci milik AWS.
- Anda dapat menambahkan atau menghapus pengguna IAM, peran IAM, dan Akun AWS dalam kebijakan utama, dan mengubah tindakan yang diizinkan atau ditolak untuk prinsipal tersebut.

Untuk informasi selengkapnya tentang cara menentukan perwakilan dan izin dalam kebijakan kunci, lihat Kebijakan utama.

- Anda tidak dapat menambahkan grup IAM ke kebijakan utama, tetapi Anda dapat menambahkan beberapa pengguna IAM dan peran IAM. Untuk informasi selengkapnya, lihat Mengizinkan beberapa prinsipal IAM untuk mengakses kunci KMS.
- Jika Anda menambahkan Akun AWS eksternal ke kebijakan kunci, Anda juga harus menggunakan kebijakan IAM di akun eksternal untuk memberikan izin kepada pengguna, grup, atau role IAM di akun tersebut. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.
- Dokumen kebijakan kunci yang dihasilkan tidak dapat melebihi 32 KB (32.768 byte).

Topik

Cara mengubah kebijakan kunci

Anda dapat mengubah kebijakan kunci dalam tiga cara berbeda seperti yang dijelaskan di bagian berikut.

Topik

Menggunakan tampilan default AWS Management Console

Anda dapat menggunakan konsol tersebut untuk mengubah kebijakan kunci dengan antarmuka grafis yang disebut tampilan default.

Jika langkah-langkah berikut tidak cocok dengan apa yang Anda lihat di konsol tersebut, mungkin berarti kebijakan kunci ini tidak dibuat oleh konsol tersebut. Atau mungkin berarti bahwa kebijakan kunci telah diubah dengan cara yang tidak didukung oleh tampilan default konsol tersebut. Dalam hal ini, ikuti langkah-langkah di Menggunakan tampilan kebijakan AWS Management Console atau Menggunakan API AWS KMS.

1. Lihat kebijakan kunci untuk kunci terkelola pelanggan seperti yang dijelaskan dalamMelihat kebijakan kunci (konsol). (Anda tidak dapat mengubah kebijakan utamaKunci yang dikelola AWS.)
2. Menentukan hal yang harus diubah.

   - Untuk menambah atau menghapus administrator kunci, dan untuk mengizinkan atau mencegah administrator kunci menghapus kunci KMS, gunakan kontrol di bagian Administrator kunci halaman. Administrator kunci mengelola kunci KMS, termasuk mengaktifkan dan menonaktifkannya, menetapkan kebijakan kunci, dan mengaktifkan rotasi kunci.
   - Untuk menambah atau menghapus pengguna kunci, dan untuk mengizinkan atau melarang eksternal Akun AWS menggunakan kunci KMS, gunakan kontrol di bagian Pengguna kunci halaman. Pengguna kunci dapat menggunakan kunci KMS dalam operasi kriptografi, seperti mengenkripsi, mendekripsi, mengenkripsi ulang, dan menghasilkan kunci data.

Menggunakan tampilan kebijakan AWS Management Console

Anda dapat menggunakan konsol tersebut untuk mengubah dokumen kebijakan kunci dengan tampilan kebijakan konsol tersebut.

1. Lihat kebijakan kunci untuk kunci terkelola pelanggan seperti yang dijelaskan dalamMelihat kebijakan kunci (konsol). (Anda tidak dapat mengubah kebijakan utamaKunci yang dikelola AWS.)
2. Di bagian Kebijakan Kunci, pilih Beralih ke tampilan kebijakan.
3. Edit dokumen kebijakan kunci, lalu pilih Simpan perubahan.

Menggunakan API AWS KMS

Anda dapat menggunakan PutKeyPolicyoperasi untuk mengubah kebijakan kunci kunci KMS di AndaAkun AWS. Anda tidak dapat menggunakan API ini pada kunci KMS yang berbedaAkun AWS.

1. Gunakan GetKeyPolicyoperasi untuk mendapatkan dokumen kebijakan kunci yang ada, lalu simpan dokumen kebijakan kunci ke file. Untuk kode sampel dalam beberapa bahasa pemrograman, lihat Mendapatkan kebijakan kunci.
2. Buka dokumen kebijakan kunci di editor teks pilihan Anda, edit dokumen kebijakan kunci, lalu simpan file.

3. Gunakan PutKeyPolicyoperasi untuk menerapkan dokumen kebijakan kunci yang diperbarui ke kunci KMS. Untuk kode sampel dalam beberapa bahasa pemrograman, lihat Mengatur kebijakan kunci.

Untuk contoh menyalin kebijakan kunci dari satu kunci KMS ke kunci lainnya, lihat GetKeyPolicy contoh di AWS CLI Command Reference.

Mengizinkan beberapa prinsipal IAM untuk mengakses kunci KMS

Grup IAM bukan perwakilan yang valid dalam kebijakan kunci. Untuk memungkinkan beberapa pengguna dan peran mengakses kunci KMS, lakukan salah satu hal berikut:

- Gunakan peran IAM sebagai prinsipal dalam kebijakan kunci. Beberapa pengguna yang berwenang dapat mengambil peran sesuai kebutuhan. Untuk detailnya, lihat peran IAM di Panduan Pengguna IAM.

  Meskipun Anda dapat mencantumkan beberapa pengguna IAM dalam kebijakan utama, praktik ini tidak disarankan karena mengharuskan Anda memperbarui kebijakan kunci setiap kali daftar pengguna yang berwenang berubah. Selain itu, praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.
- Gunakan kebijakan IAM untuk memberikan izin kepada grup IAM. Untuk melakukan ini, pastikan bahwa kebijakan kunci mencakup pernyataan yang memungkinkan kebijakan IAM untuk mengizinkan akses ke kunci KMS, membuat kebijakan IAM yang memungkinkan akses ke kunci KMS, dan kemudian melampirkan kebijakan itu ke grup IAM yang berisi pengguna IAM yang berwenang. Dengan menggunakan pendekatan ini, Anda tidak perlu mengubah kebijakan apa pun ketika daftar pengguna yang diotorisasi berubah. Sebaliknya, Anda hanya perlu menambahkan atau menghapus pengguna tersebut dari grup IAM yang sesuai. Untuk detailnya, lihat grup pengguna IAM di Panduan Pengguna IAM

Untuk informasi selengkapnya cara kerja kebijakan kunci AWS KMS dan kebijakan IAM, lihat Memecahkan masalah akses kunci.

Izin untuk AWS layanan dalam kebijakan utama

Banyak AWS layanan digunakan AWS KMS keys untuk melindungi sumber daya yang mereka kelola. Ketika suatu layanan menggunakan Kunci milik AWSatau Kunci yang dikelola AWS, layanan menetapkan dan memelihara kebijakan utama untuk kunci KMS ini.

Namun, ketika Anda menggunakan kunci yang dikelola pelanggan dengan AWS layanan, Anda menetapkan dan mempertahankan kebijakan kunci. Kebijakan utama tersebut harus mengizinkan layanan izin minimum yang diperlukan untuk melindungi sumber daya atas nama Anda. Kami menyarankan Anda mengikuti prinsip hak istimewa paling sedikit: berikan layanan hanya izin yang diperlukan. Anda dapat melakukannya secara efektif dengan mempelajari izin mana yang dibutuhkan layanan dan menggunakan kunci kondisi AWS global dan kunci AWS KMS kondisi untuk menyempurnakan izin.

Untuk menemukan izin yang diperlukan layanan pada kunci yang dikelola pelanggan, lihat dokumentasi enkripsi untuk layanan tersebut. Misalnya, untuk izin yang diperlukan Amazon Elastic Block Store (Amazon EBS), lihat Izin untuk pengguna IAM di Panduan Pengguna Amazon EC2 untukInstans Linux dan Panduan Pengguna Amazon EC2 untuk Instans Windows. Untuk izin yang dibutuhkan Secrets Manager, lihat Mengotorisasi penggunaan kunci KMS di Panduan Pengguna.AWS Secrets Manager

Menerapkan izin yang paling tidak diistimewakan

Ketika Anda memberikan izin AWS layanan untuk menggunakan kunci KMS, pastikan bahwa izin tersebut hanya berlaku untuk sumber daya yang harus diakses layanan atas nama Anda. Strategi hak istimewa terkecil ini membantu mencegah penggunaan kunci KMS yang tidak sah saat permintaan diteruskan antar layanan. AWS

Untuk menerapkan strategi hak istimewa terkecil, gunakan kami sarankan menggunakan kunci kondisi konteks AWS KMS enkripsi dan ARN sumber global atau kunci kondisi akun sumber.

Menggunakan kunci kondisi konteks enkripsi

Cara paling efektif untuk menerapkan izin yang paling tidak memiliki hak istimewa saat menggunakan AWS KMS sumber daya adalah dengan memasukkan kms:EncryptionContext:context-keyatau kunci kms:EncryptionContextKeyskondisi dalam kebijakan yang memungkinkan kepala sekolah memanggil operasi kriptografi. AWS KMS Kunci kondisi ini sangat efektif karena mengaitkan izin dengan konteks enkripsi yang terikat pada ciphertext saat sumber daya dienkripsi.

Gunakan kunci kondisi konteks enkripsi hanya jika tindakan dalam pernyataan kebijakan adalah CreateGrantatau operasi kriptografi AWS KMS simetris yang mengambil `EncryptionContext` parameter, seperti operasi seperti GenerateDataKeyatau Dekripsi. (Untuk daftar operasi yang didukung, lihat kms:EncryptionContext:context-keyatau kms:EncryptionContextKeys.) Jika Anda menggunakan kunci kondisi ini untuk mengizinkan operasi lain, seperti DescribeKey, izin akan ditolak.

Tetapkan nilai ke konteks enkripsi yang digunakan layanan saat mengenkripsi sumber daya. Informasi ini biasanya tersedia di bagian Keamanan dokumentasi layanan. Misalnya, konteks enkripsi untuk AWS Proton mengidentifikasi sumber daya Proton dan template AWS terkaitnya. Konteks AWS Secrets Manager enkripsi mengidentifikasi rahasia dan versinya. Konteks enkripsi untuk Lokasi Amazon mengidentifikasi pelacak atau koleksi.

Contoh pernyataan kebijakan kunci berikut memungkinkan Amazon Location Service untuk membuat hibah atas nama pengguna yang berwenang. Pernyataan kebijakan ini membatasi izin dengan menggunakan kms: ViaService, kms: CallerAccount, dan kunci `kms:EncryptionContext:`*`context-key`* kondisi untuk mengikat izin ke sumber daya pelacak tertentu.

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

Menggunakan **`aws:SourceArn`** atau **`aws:SourceAccount`** mengkondisikan kunci

Ketika prinsipal dalam pernyataan kebijakan kunci adalah prinsip AWS layanan, kami sangat menyarankan Anda menggunakan aws:SourceArnatau kunci kondisi aws:SourceAccountglobal, selain kunci `kms:EncryptionContext:`*`context-key`* kondisi. ARN dan nilai akun disertakan dalam konteks otorisasi hanya ketika permintaan datang AWS KMS dari layanan lain. AWS Kombinasi kondisi ini menerapkan izin yang paling tidak memiliki hak istimewa dan menghindari skenario wakil yang berpotensi membingungkan. Prinsipal layanan biasanya tidak digunakan sebagai prinsipal dalam kebijakan utama, tetapi beberapa AWS layanan, seperti, memerlukannya. AWS CloudTrail

Untuk menggunakan `aws:SourceArn` atau kunci kondisi `aws:SourceAccount` global, tetapkan nilai ke Amazon Resource Name (ARN) atau akun sumber daya yang sedang dienkripsi. Misalnya, dalam pernyataan kebijakan kunci yang memberikan AWS CloudTrail izin untuk mengenkripsi jejak, tetapkan nilai `aws:SourceArn` ke ARN jejak. Bila memungkinkan, gunakan`aws:SourceArn`, yang lebih spesifik. Tetapkan nilai ke ARN atau pola ARN dengan karakter wildcard. Jika Anda tidak tahu ARN sumber daya, gunakan `aws:SourceAccount` sebagai gantinya.

> **Note**
>
> Jika ARN sumber daya menyertakan karakter yang tidak diizinkan dalam kebijakan AWS KMS kunci, Anda tidak dapat menggunakan ARN sumber daya tersebut dalam nilai kunci kondisi. `aws:SourceArn` Sebagai gantinya, gunakan tombol `aws:SourceAccount` kondisi. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama.

Dalam contoh kebijakan kunci berikut, prinsipal yang mendapatkan izin adalah prinsip AWS CloudTrail layanan,`cloudtrail.amazonaws.com`. Untuk menerapkan hak istimewa paling sedikit, kebijakan ini menggunakan kunci `aws:SourceArn` dan `kms:EncryptionContext:`*`context-key`* kondisi. Pernyataan kebijakan memungkinkan CloudTrail untuk menggunakan kunci KMS untuk menghasilkan kunci data yang digunakan untuk mengenkripsi jejak. `kms:EncryptionContext:`*`context-key`*Kondisi `aws:SourceArn` dan dievaluasi secara independen. Setiap permintaan untuk menggunakan kunci KMS untuk operasi yang ditentukan harus memenuhi kedua kondisi.

Untuk membatasi izin layanan ke `finance` jejak di akun contoh (111122223333) dan `us-west-2` Wilayah, pernyataan kebijakan ini menetapkan `aws:SourceArn` kunci kondisi ke ARN dari jejak tertentu. Pernyataan kondisi menggunakan ArnEqualsoperator untuk memastikan bahwa setiap elemen dalam ARN dievaluasi secara independen saat mencocokkan. Contoh ini juga menggunakan kunci `kms:EncryptionContext:`*`context-key`* kondisi untuk membatasi izin untuk melacak di akun dan Wilayah tertentu.

Sebelum menggunakan kebijakan kunci ini, ganti contoh ID akun, Wilayah, dan nama jejak dengan nilai yang valid dari akun Anda.

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

Menggunakan kebijakan IAM dengan AWS KMS

Anda dapat menggunakan kebijakan IAM, bersama dengan kebijakan utama, hibah, dan kebijakan titik akhir VPC, untuk mengontrol akses ke akun Anda. AWS KMS keys AWS KMS

> **Note**
>
> Untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS, kebijakan kunci untuk kunci KMS harus memberikan izin akun untuk menggunakan kebijakan IAM. Khususnya, kebijakan kunci harus menyertakan pernyataan kebijakan yang mengaktifkan kebijakan IAM.
>
> Bagian ini menjelaskan cara menggunakan kebijakan IAM untuk mengontrol akses ke AWS KMS operasi. Untuk informasi umum tentang IAM, lihat Panduan Pengguna IAM.

Semua kunci KMS harus memiliki kebijakan kunci. Kebijakan IAM bersifat opsional. Untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS, kebijakan kunci untuk kunci

KMS harus memberikan izin akun untuk menggunakan kebijakan IAM. Khususnya, kebijakan kunci harus menyertakan pernyataan kebijakan yang mengaktifkan kebijakan IAM.

Kebijakan IAM dapat mengontrol akses ke AWS KMS operasi apa pun. Tidak seperti kebijakan utama, kebijakan IAM dapat mengontrol akses ke beberapa kunci KMS dan memberikan izin untuk operasi beberapa layanan terkait. AWS Tetapi kebijakan IAM sangat berguna untuk mengendalikan akses ke operasi, seperti CreateKey, yang tidak dapat dikendalikan oleh kebijakan utama karena mereka tidak melibatkan kunci KMS tertentu.

Jika Anda mengakses AWS KMS melalui titik akhir Amazon Virtual Private Cloud (Amazon VPC), Anda juga dapat menggunakan kebijakan titik akhir VPC untuk membatasi akses ke sumber daya saat menggunakan titik akhir. AWS KMS Misalnya, saat menggunakan titik akhir VPC, Anda mungkin hanya mengizinkan prinsipal di Anda Akun AWS untuk mengakses kunci terkelola pelanggan Anda. Untuk detailnya, lihat Mengontrol akses ke VPC endpoint.

Untuk mendapatkan bantuan mengenai cara menulis dan memformat dokumen kebijakan JSON, lihat Referensi Kebijakan IAM JSON dalam Panduan Pengguna IAM.

Topik

Gambaran umum dari kebijakan IAM

Anda dapat menggunakan kebijakan IAM dengan cara berikut:

- Melampirkan kebijakan izin ke peran untuk izin federasi atau lintas akun – Anda dapat melampirkan kebijakan IAM untuk IAM role untuk mengaktifkan federasi identitas, mengizinkan izin lintas-akun, atau memberikan izin untuk aplikasi yang berjalan pada instans EC2. Untuk informasi selengkapnya tentang berbagai kasus penggunaan IAM role, lihat IAM Role dalam Panduan Pengguna IAM.
- Melampirkan kebijakan izin ke pengguna atau grup — Anda dapat melampirkan kebijakan yang memungkinkan pengguna atau grup pengguna untuk memanggil AWS KMS operasi. Namun,

praktik terbaik IAM merekomendasikan agar Anda menggunakan identitas dengan kredensi sementara, seperti peran IAM, bila memungkinkan.

Contoh berikut menunjukkan kebijakan IAM dengan AWS KMS izin. Kebijakan ini memungkinkan identitas IAM yang dilampirkan untuk mencantumkan semua kunci dan alias KMS.

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

Seperti semua kebijakan IAM, kebijakan ini tidak memiliki elemen `Principal`. Saat Anda melampirkan kebijakan IAM ke identitas IAM, identitas tersebut akan mendapatkan izin yang ditentukan dalam kebijakan tersebut.

Untuk tabel yang menunjukkan semua tindakan AWS KMS API dan sumber daya yang diterapkan, lihatReferensi izin.

Praktik terbaik untuk kebijakan IAM

Mengamankan akses ke AWS KMS keys sangat penting untuk keamanan semua sumber AWS daya Anda. Kunci KMS digunakan untuk melindungi banyak sumber daya paling sensitif di Anda Akun AWS. Luangkan waktu untuk merancang kebijakan utama, kebijakan IAM, hibah, dan kebijakan titik akhir VPC yang mengontrol akses ke kunci KMS Anda.

Dalam pernyataan kebijakan IAM yang mengontrol akses ke kunci KMS, gunakan prinsip yang paling tidak memiliki hak istimewa. Berikan kepala sekolah IAM hanya izin yang mereka butuhkan hanya pada kunci KMS yang harus mereka gunakan atau kelola.

Praktik terbaik berikut berlaku untuk kebijakan IAM yang mengontrol akses ke AWS KMS kunci dan alias. Untuk panduan praktik terbaik kebijakan IAM umum, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Gunakan kebijakan kunci

Jika memungkinkan, berikan izin dalam kebijakan utama yang memengaruhi satu kunci KMS, bukan dalam kebijakan IAM yang dapat diterapkan ke banyak kunci KMS, termasuk yang lain. Akun AWS Ini sangat penting untuk izin sensitif seperti kms: PutKeyPolicy dan kms: ScheduleKeyDeletion tetapi juga untuk operasi kriptografi yang menentukan bagaimana data Anda dilindungi.

Batasi CreateKey izin

Berikan izin untuk membuat kunci (kms: CreateKey) hanya untuk kepala sekolah yang membutuhkannya. Prinsipal yang membuat kunci KMS juga menetapkan kebijakan utamanya, sehingga mereka dapat memberi diri mereka sendiri dan orang lain izin untuk menggunakan dan mengelola kunci KMS yang mereka buat. Jika Anda mengizinkan izin ini, pertimbangkan untuk membatasinya menggunakan syarat kebijakan. Misalnya, Anda dapat menggunakan KeySpec kondisi kms: untuk membatasi izin ke kunci KMS enkripsi simetris.

Tentukan kunci KMS dalam kebijakan IAM

Sebagai praktik terbaik, tentukan ARN kunci dari setiap kunci KMS yang izinnya berlaku dalam `Resource` elemen pernyataan kebijakan. Praktik ini membatasi izin ke kunci KMS yang dibutuhkan kepala sekolah. Misalnya, `Resource` elemen ini hanya mencantumkan kunci KMS yang perlu digunakan oleh prinsipal.

```
"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"
]
```

Saat menentukan kunci KMS tidak praktis, gunakan `Resource` nilai yang membatasi akses ke kunci KMS di tepercaya Akun AWS dan Wilayah, seperti. `arn:aws:kms:`*`region`*`:`*`account`*`:key/*` Atau batasi akses ke kunci KMS di semua Wilayah (*) yang terpercaya Akun AWS, seperti`arn:aws:kms:*:`*`account`*`:key/*`.

Anda tidak dapat menggunakan ID kunci, nama alias, atau alias ARN untuk mewakili kunci KMS di `Resource` bidang kebijakan IAM. Jika Anda menentukan alias ARN, kebijakan berlaku untuk alias, bukan ke kunci KMS. Untuk informasi tentang kebijakan IAM alias, lihat Mengontrol akses ke alias

Hindari "Sumber Daya": "*" dalam kebijakan IAM

Gunakan karakter kartubebas (*) dengan bijaksana. Dalam kebijakan kunci, karakter wildcard dalam `Resource` elemen mewakili kunci KMS tempat kebijakan kunci dilampirkan. Namun dalam kebijakan IAM, karakter wildcard saja di `Resource` element (`"Resource": "*"`) menerapkan izin ke semua kunci KMS di semua Akun AWS yang diizinkan oleh akun prinsipal untuk digunakan. Ini mungkin termasuk kunci KMS di lain Akun AWS, serta kunci KMS di akun kepala sekolah.

Misalnya, untuk menggunakan kunci KMS di akun lain Akun AWS, kepala sekolah memerlukan izin dari kebijakan kunci KMS di akun eksternal, dan dari kebijakan IAM di akun mereka sendiri. Misalkan akun arbitrer memberikan izin Akun AWS KMS: Decrypt Anda pada kunci KMS mereka. Jika demikian, kebijakan IAM di akun Anda yang memberikan `kms:Decrypt` izin peran pada semua kunci KMS (`"Resource": "*"`) akan memenuhi bagian IAM dari persyaratan tersebut. Akibatnya, kepala sekolah yang dapat mengasumsikan peran itu sekarang dapat mendekripsi ciphertext menggunakan kunci KMS di akun yang tidak tepercaya. Entri untuk operasi mereka muncul di CloudTrail log kedua akun.

Secara khusus, hindari penggunaan `"Resource": "*"` dalam pernyataan kebijakan yang mengizinkan operasi API berikut. Operasi ini dapat dipanggil pada kunci KMS di lain Akun AWS.

- DescribeKey
- GetKeyRotationStatus
- Operasi kriptografi (Enkripsi, Dekripsi,,,, GenerateDataKey, GenerateDataKeyPair, Tanda GenerateDataKeyWithoutPlaintextGenerateDataKeyPairWithoutPlaintextGetPublicKeyReEncrypt, Verifikasi)
- CreateGrant, ListGrants, ListRetirableGrants, RetireGrant, RevokeGrant

Waktu menggunakan "Sumber Daya": "*"

Dalam kebijakan IAM, gunakan karakter kartubebas di elemen `Resource` hanya untuk izin yang memerlukannya. Hanya izin berikut yang memerlukan elemen `"Resource": "*"`.

- km: CreateKey
- km: GenerateRandom
- km: ListAliases
- km: ListKeys
- Izin untuk toko kunci khusus, seperti kms: CreateCustomKeyStore dan kms:. ConnectCustomKeyStore

> **Note**
>
> Izin untuk operasi alias (kms: CreateAlias, kms:UpdateAlias, kms: DeleteAlias) harus dilampirkan ke alias dan kunci KMS. Anda dapat menggunakan `"Resource": "*"` dalam kebijakan IAM untuk mewakili alias dan kunci KMS, atau menentukan alias dan kunci KMS dalam elemen. `Resource` Sebagai contoh, lihat Mengontrol akses ke alias.

Contoh dalam topik ini memberikan lebih banyak informasi dan panduan untuk merancang kebijakan IAM untuk kunci KMS. Untuk panduan praktik AWS KMS terbaik umum, lihat Praktik AWS Key Management Service Terbaik (PDF). Untuk praktik terbaik IAM untuk semua AWS sumber daya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Menentukan kunci KMS dalam pernyataan kebijakan IAM

Anda dapat menggunakan kebijakan IAM untuk mengizinkan prinsipal menggunakan atau mengelola kunci KMS. Kunci KMS ditentukan dalam `Resource` elemen pernyataan kebijakan.

- Untuk menentukan kunci KMS dalam pernyataan kebijakan IAM, Anda harus menggunakan kunci ARN. Anda tidak dapat menggunakan ID kunci, nama alias, atau alias ARN untuk mengidentifikasi kunci KMS dalam pernyataan kebijakan IAM.

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

  Untuk mengontrol akses ke kunci KMS berdasarkan aliasnya, gunakan kunci kondisi kms: RequestAlias atau kms:. ResourceAliases Lihat perinciannya di ABAC untuk AWS KMS.

  Gunakan alias ARN sebagai sumber daya hanya dalam pernyataan kebijakan yang mengontrol akses ke operasi alias, CreateAliasseperti UpdateAlias,, atau. DeleteAlias Lihat perinciannya di Mengontrol akses ke alias.
- Untuk menentukan beberapa kunci KMS di akun dan Wilayah, gunakan karakter wildcard (*) di wilayah atau posisi ID sumber daya ARN kunci.

  Misalnya, untuk menentukan semua kunci KMS di Wilayah AS Barat (Oregon) akun, gunakan `""Resource": "arn:aws:kms:us-west-2:111122223333:key/*`.

Untuk menentukan semua kunci KMS di semua Wilayah akun, gunakan "`Resource": "arn:aws:kms:*:111122223333:key/*`".

- Untuk mewakili semua kunci KMS, gunakan karakter wildcard alone ()"`*`". Gunakan format ini untuk operasi yang tidak menggunakan kunci KMS tertentu, yaitu, CreateKey, GenerateRandomListAliases, dan ListKeys.

Saat menulis pernyataan kebijakan Anda, sebaiknya tentukan hanya kunci KMS yang perlu digunakan oleh prinsipal, daripada memberi mereka akses ke semua kunci KMS.

Misalnya, pernyataan kebijakan IAM berikut memungkinkan prinsipal untuk memanggil DescribeKey, GenerateDataKey, Dekripsi operasi hanya pada kunci KMS yang tercantum dalam `Resource` elemen pernyataan kebijakan. Menentukan kunci KMS dengan kunci ARN, yang merupakan praktik terbaik, memastikan bahwa izin hanya terbatas pada kunci KMS yang ditentukan.

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

Untuk menerapkan izin ke semua kunci KMS di tepercaya tertentu Akun AWS, Anda dapat menggunakan karakter wildcard (*) di posisi Region dan ID kunci. Misalnya, pernyataan kebijakan berikut memungkinkan prinsipal untuk memanggil operasi yang ditentukan pada semua kunci KMS dalam dua akun contoh tepercaya.

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

Anda juga dapat menggunakan karakter kartubebas (`"*"`) saja di elemen `Resource`. Karena memungkinkan akses ke semua kunci KMS akun memiliki izin untuk digunakan, disarankan terutama untuk operasi tanpa kunci KMS tertentu dan untuk `Deny` pernyataan. Anda juga dapat menggunakannya dalam pernyataan kebijakan yang mengizinkan operasi hanya baca yang sensitif. Untuk menentukan apakah AWS KMS operasi melibatkan kunci KMS tertentu, cari nilai kunci KMS di kolom Sumber daya tabel di. the section called “Referensi izin”

Misalnya, pernyataan kebijakan berikut menggunakan `Deny` efek untuk melarang prinsipal menggunakan operasi tertentu pada kunci KMS apa pun. Ini menggunakan karakter wildcard dalam `Resource` elemen untuk mewakili semua kunci KMS.

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

Pernyataan kebijakan berikut menggunakan karakter wildcard saja untuk mewakili semua kunci KMS. Tetapi ini hanya memungkinkan operasi dan operasi hanya-baca yang kurang sensitif yang tidak berlaku untuk kunci KMS tertentu.

```
{
```

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

Izin yang diperlukan untuk menggunakan konsol AWS KMS

Untuk bekerja dengan AWS KMS konsol, pengguna harus memiliki set izin minimum yang memungkinkan mereka untuk bekerja dengan AWS KMS sumber daya di dalamnya. Akun AWS Selain AWS KMS izin ini, pengguna juga harus memiliki izin untuk mencantumkan pengguna IAM dan peran IAM. Jika Anda membuat kebijakan IAM yang lebih ketat daripada izin minimum yang diperlukan, AWS KMS konsol tidak akan berfungsi seperti yang dimaksudkan untuk pengguna dengan kebijakan IAM tersebut.

Untuk izin minimum yang diperlukan guna mengizinkan pengguna akses hanya baca ke konsol AWS KMS tersebut, lihat Izinkan pengguna untuk melihat kunci KMS di konsol AWS KMS.

Untuk memungkinkan pengguna bekerja dengan AWS KMS konsol untuk membuat dan mengelola kunci KMS, lampirkan kebijakan AWSKeyManagementServicePowerUserterkelola ke pengguna, seperti yang dijelaskan di bagian berikut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang bekerja dengan AWS KMS API melalui AWS SDK, AWS Command Line Interfaceatau. AWS Tools for PowerShell Namun, Anda harus memberikan izin kepada pengguna ini untuk menggunakan API. Untuk informasi selengkapnya, lihat Referensi izin.

AWS kebijakan terkelola untuk pengguna listrik

Anda dapat menggunakan kebijakan `AWSKeyManagementServicePowerUser` terkelola untuk memberikan izin kepada pengguna daya kepada kepala IAM di akun Anda. Pengguna daya dapat membuat kunci KMS, menggunakan dan mengelola kunci KMS yang mereka buat, dan melihat semua kunci KMS dan identitas IAM. Prinsipal yang memiliki kebijakan

`AWSKeyManagementServicePowerUser` terkelola juga bisa mendapatkan izin dari sumber lain, termasuk kebijakan utama, kebijakan IAM lainnya, dan hibah.

`AWSKeyManagementServicePowerUser`adalah kebijakan IAM AWS terkelola. Untuk informasi selengkapnya tentang kebijakan AWS AWS terkelola, lihat kebijakan terkelola di Panduan Pengguna IAM.

> **Note**
>
> Izin dalam kebijakan ini yang khusus untuk kunci KMS, seperti `kms:TagResource` dan`kms:GetKeyRotationStatus`, hanya efektif jika kebijakan utama untuk kunci KMS tersebut secara eksplisit mengizinkan kebijakan IAM untuk menggunakan kebijakan IAM Akun AWS untuk mengontrol akses ke kunci tersebut. Untuk menentukan apakah izin khusus untuk kunci KMS, lihat AWS KMS izin dan cari nilai kunci KMS di kolom Sumber Daya. Kebijakan ini memberikan izin pengguna daya pada kunci KMS apa pun dengan kebijakan kunci yang mengizinkan pengoperasian. Untuk izin lintas akun, seperti `kms:DescribeKey` dan`kms:ListGrants`, ini mungkin termasuk kunci KMS yang tidak dipercaya. Akun AWS Untuk detailnya, lihat Praktik terbaik untuk kebijakan IAM dan Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS. Untuk menentukan apakah izin valid pada kunci KMS di akun lain, lihat AWS KMS izin dan cari nilai Ya di kolom Penggunaan lintas akun. Untuk mengizinkan prinsipal melihat AWS KMS konsol tanpa kesalahan, prinsipal memerlukan GetResources izin tag:, yang tidak termasuk dalam kebijakan. `AWSKeyManagementServicePowerUser` Anda dapat mengizinkan izin ini dalam kebijakan IAM terpisah.

Kebijakan IAM AWSKeyManagementServicePowerUserterkelola mencakup izin berikut.

- Memungkinkan kepala sekolah untuk membuat kunci KMS. Karena proses ini mencakup pengaturan kebijakan kunci, pengguna daya dapat memberikan izin kepada diri mereka sendiri dan orang lain untuk menggunakan dan mengelola kunci KMS yang mereka buat.
- Memungkinkan prinsipal untuk membuat dan menghapus alias dan tag pada semua kunci KMS. Mengubah tag atau alias dapat mengizinkan atau menolak izin untuk menggunakan dan mengelola kunci KMS. Lihat perinciannya di ABAC untuk AWS KMS.
- Memungkinkan prinsipal untuk mendapatkan informasi rinci tentang semua kunci KMS, termasuk ARN kunci mereka, konfigurasi kriptografi, kebijakan kunci, alias, tag, dan status rotasi.
- Memungkinkan prinsipal untuk mencantumkan pengguna, grup, dan peran IAM.

- Kebijakan ini tidak mengizinkan prinsipal untuk menggunakan atau mengelola kunci KMS yang tidak mereka buat. Namun, mereka dapat mengubah alias dan tag pada semua kunci KMS, yang mungkin mengizinkan atau menolak izin mereka untuk menggunakan atau mengelola kunci KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateAlias",
                "kms:CreateKey",
                "kms:DeleteAlias",
                "kms:Describe*",
                "kms:GenerateRandom",
                "kms:Get*",
                "kms:List*",
                "kms:TagResource",
                "kms:UntagResource",
                "iam:ListGroups",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Contoh kebijakan IAM

Dalam bagian ini, Anda dapat menemukan contoh kebijakan IAM yang mengizinkan izin untuk berbagai tindakan AWS KMS .

> **Important**
>
> Beberapa izin dalam kebijakan berikut hanya diizinkan jika kebijakan kunci KMS juga mengizinkannya. Untuk informasi selengkapnya, lihat Referensi izin.

Untuk mendapatkan bantuan mengenai cara menulis dan memformat dokumen kebijakan JSON, lihat Referensi Kebijakan IAM JSON dalam Panduan Pengguna IAM.

Contoh

- Izinkan pengguna untuk melihat kunci KMS di konsol AWS KMS
- Izinkan pengguna untuk membuat kunci KMS
- Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS apa pun secara spesifik Akun AWS
- Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS apa pun di spesifik dan Wilayah Akun AWS
- Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS tertentu
- Mencegah pengguna menonaktifkan atau menghapus tombol KMS

Izinkan pengguna untuk melihat kunci KMS di konsol AWS KMS

Kebijakan IAM berikut memungkinkan pengguna akses hanya-baca ke konsol. AWS KMS Pengguna dengan izin ini dapat melihat semua kunci KMS di dalamnya Akun AWS, tetapi mereka tidak dapat membuat atau mengubah kunci KMS apa pun.

Untuk melihat kunci KMS pada halaman kunci yang dikelola Pelanggan Kunci yang dikelola AWSdan pelanggan, prinsipal memerlukan GetResources izin kms: ListKeys, kms:, dan tag: ListAliases, meskipun kunci tidak memiliki tag atau alias. Izin yang tersisa, terutama kms: DescribeKey, diperlukan untuk melihat kolom tabel kunci KMS opsional dan data pada halaman detail kunci KMS. ListRolesIzin iam: ListUsers dan iam: diperlukan untuk menampilkan kebijakan kunci dalam tampilan default tanpa kesalahan. Untuk melihat data di halaman toko kunci Kustom dan detail tentang kunci KMS di toko kunci khusus, kepala sekolah juga memerlukan izin kms:. DescribeCustomKeyStores

Jika Anda membatasi akses konsol pengguna ke kunci KMS tertentu, konsol akan menampilkan kesalahan untuk setiap kunci KMS yang tidak terlihat.

Kebijakan ini mencakup dua pernyataan kebijakan. `Resource`Elemen dalam pernyataan kebijakan pertama memungkinkan izin yang ditentukan pada semua kunci KMS di semua Wilayah contoh. Akun AWS Pemirsa konsol tidak memerlukan akses tambahan karena AWS KMS konsol hanya menampilkan kunci KMS di akun prinsipal. Ini benar bahkan jika mereka memiliki izin untuk melihat kunci KMS di lain Akun AWS. Izin yang tersisa AWS KMS dan IAM memerlukan `"Resource": "*"` elemen karena tidak berlaku untuk kunci KMS tertentu.

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

Izinkan pengguna untuk membuat kunci KMS

Kebijakan IAM berikut memungkinkan pengguna untuk membuat semua jenis kunci KMS. Nilai `Resource` elemen adalah `*` karena `CreateKey` operasi tidak menggunakan AWS KMS sumber daya tertentu (kunci KMS atau alias).

Untuk membatasi pengguna pada jenis kunci KMS tertentu, gunakan kunci kondisi kms:, kms: KeySpecKeyUsage, dan kms:. KeyOrigin

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

```
  "Statement": {
    "Effect": "Allow",
    "Action": "kms:CreateKey",
    "Resource": "*"
  }
}
```

Perwakilan yang membuat kunci mungkin memerlukan beberapa izin terkait.

- kms: PutKeyPolicy — Kepala sekolah yang memiliki `kms:CreateKey` izin dapat mengatur kebijakan kunci awal untuk kunci KMS. Namun, `CreateKey` pemanggil harus memiliki PutKeyPolicy izin kms:, yang memungkinkan mereka mengubah kebijakan kunci KMS, atau mereka harus menentukan `BypassPolicyLockoutSafetyCheck` parameter`CreateKey`, yang tidak disarankan. `CreateKey`Penelepon bisa mendapatkan `kms:PutKeyPolicy` izin untuk kunci KMS dari kebijakan IAM atau mereka dapat menyertakan izin ini dalam kebijakan kunci kunci KMS yang mereka buat.
- kms: TagResource — Untuk menambahkan tag ke kunci KMS selama `CreateKey` operasi, `CreateKey` penelepon harus memiliki TagResource izin kms: dalam kebijakan IAM. Menyertakan izin ini dalam kebijakan kunci KMS baru tidak cukup. Namun, jika `CreateKey` pemanggil menyertakan `kms:TagResource` dalam kebijakan kunci awal, mereka dapat menambahkan tag dalam panggilan terpisah setelah kunci KMS dibuat.
- kms: CreateAlias — Prinsipal yang membuat kunci KMS di AWS KMS konsol harus memiliki CreateAlias izin kms: pada kunci KMS dan pada alias. (Konsol tersebut membuat dua panggilan; ke `CreateKey` dan ke `CreateAlias`). Anda harus memberikan izin alias dalam kebijakan IAM. Anda dapat memberikan izin kunci KMS dalam kebijakan utama atau kebijakan IAM. Lihat perinciannya di Mengontrol akses ke alias.

Selain itu`kms:CreateKey`, kebijakan IAM berikut memberikan `kms:TagResource` izin pada semua kunci KMS di Akun AWS dan `kms:CreateAlias` izin pada semua alias yang akun. Ini juga menyertakan beberapa izin hanya baca berguna yang dapat diberikan hanya dalam kebijakan IAM.

Kebijakan IAM ini tidak menyertakan izin `kms:PutKeyPolicy` atau izin lain yang dapat diatur dalam kebijakan kunci. Merupakan praktik terbaik untuk menetapkan izin ini dalam kebijakan utama di mana izin tersebut berlaku secara eksklusif ke satu kunci KMS.

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

Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS apa pun secara spesifik Akun AWS

Kebijakan IAM berikut memungkinkan pengguna untuk mengenkripsi dan mendekripsi data dengan kunci KMS apa pun di 111122223333. Akun AWS

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

Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS apa pun di spesifik dan Wilayah Akun AWS

Kebijakan IAM berikut memungkinkan pengguna untuk mengenkripsi dan mendekripsi data dengan kunci KMS apa pun Akun AWS 111122223333 di Wilayah AS Barat (Oregon).

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

Izinkan pengguna untuk mengenkripsi dan mendekripsi dengan kunci KMS tertentu

Kebijakan IAM berikut memungkinkan pengguna untuk mengenkripsi dan mendekripsi data dengan dua kunci KMS yang ditentukan dalam elemen. `Resource` Saat menentukan kunci KMS dalam pernyataan kebijakan IAM, Anda harus menggunakan kunci ARN dari kunci KMS.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

Mencegah pengguna menonaktifkan atau menghapus tombol KMS

Kebijakan IAM berikut mencegah pengguna menonaktifkan atau menghapus kunci KMS apa pun, bahkan ketika kebijakan IAM lain atau kebijakan kunci mengizinkan izin ini. Kebijakan yang secara eksplisit menolak izin menimpa semua kebijakan lain, meski kebijakan yang secara eksplisit memungkinkan izin yang sama. Untuk informasi selengkapnya, lihat Memecahkan masalah akses kunci.

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

Hibah di AWS KMS

Hibah adalah instrumen kebijakan yang memungkinkan AWSprinsipal untuk menggunakan kunci KMS dalam operasi kriptografi. Hal ini juga dapat membiarkan mereka melihat kunci KMS (`DescribeKey`) dan membuat dan mengelola hibah. Saat mengotorisasi akses ke kunci KMS, hibah dipertimbangkan bersama dengan kebijakan utama dan kebijakan IAM. Pemberian izin sering digunakan untuk izin sementara karena Anda dapat membuatnya, menggunakan izinnya, dan menghapusnya tanpa mengubah kebijakan kunci atau kebijakan IAM.

Pemberian izin biasanya digunakan oleh layanan AWS yang terintegrasi dengan AWS KMS untuk mengenkripsi data Anda saat tidak digunakan. Layanan akan membuat pemberian izin atas nama pengguna di akun, menggunakan izinnya, dan menghentikan pemberian izin begitu tugasnya selesai. Untuk detail tentang cara layanan AWS menggunakan pemberian izin, lihat Bagaimana layanan AWS menggunakan AWS KMS atau topik Enkripsi saat tidak digunakan di panduan pengguna layanan atau panduan developer.

Untuk contoh kode yang mendemonstrasikan pembuatan pemberian izin dalam beberapa bahasa pemrograman lihat Bekerja dengan izin.

Topik

Tentang pemberian izin

Pemberian izin adalah mekanisme kontrol akses yang sangat fleksibel dan berguna. Saat Anda membuat hibah untuk kunci KMS, hibah memungkinkan kepala penerima hibah untuk memanggil operasi hibah yang ditentukan pada kunci KMS asalkan semua kondisi yang ditentukan dalam hibah terpenuhi.

- Setiap hibah memungkinkan akses ke tepat satu kunci KMS. Anda dapat membuat hibah untuk kunci KMS yang berbedaAkun AWS.
- Hibah dapat memungkinkan akses ke kunci KMS, tetapi tidak menolak akses.
- Setiap hibah memiliki satu pokok hibah. Prinsipal penerima hibah dapat mewakili satu atau lebih identitas yang Akun AWS sama dengan kunci KMS atau dalam akun yang berbeda.
- Hibah hanya dapat memungkinkan operasi hibah. Operasi hibah harus didukung oleh kunci KMS dalam hibah. Jika Anda menentukan operasi yang tidak didukung, CreateGrantpermintaan gagal dengan `ValidationError` pengecualian.
- Prinsipal penerima hibah dapat menggunakan izin yang diberikan hibah kepada mereka tanpa menentukan hibah, sama seperti jika izin berasal dari kebijakan utama atau kebijakan IAM. Namun, karena AWS KMS API mengikuti model konsistensi akhirnya, saat Anda membuat, menghentikan, atau mencabut hibah, mungkin ada penundaan singkat, sebelum perubahan tersedia secara keseluruhan. AWS KMS Untuk segera menggunakan izin dalam pemberian izin, gunakan token izin.
- Kepala sekolah yang berwenang dapat menghapus hibah (pensiun atau mencabutnya). Menghapus hibah menghilangkan semua izin yang diizinkan oleh hibah. Anda tidak perlu mencari tahu kebijakan mana yang akan ditambahkan atau dihapus untuk membatalkan hibah.
- AWS KMSmembatasi jumlah hibah pada setiap kunci KMS. Untuk detailnya, lihat Hibah per kunci KMS: 50.000.

Berhati-hatilah saat membuat pemberian izin dan saat memberi izin kepada orang lain untuk membuat pemberian izin. Izin untuk membuat hibah memiliki implikasi keamanan, seperti mengizinkan PutKeyPolicy izin kms: untuk menetapkan kebijakan.

- Pengguna dengan izin untuk membuat hibah untuk kunci KMS (`kms:CreateGrant`) dapat menggunakan hibah untuk memungkinkan pengguna dan peran, termasuk AWS layanan, untuk menggunakan kunci KMS. Perwakilan dapat berupa identitas dalam Akun AWS atau identitas Anda sendiri di akun atau organisasi yang berbeda.
- Pemberian izin hanya dapat mengizinkan subset operasi AWS KMS. Anda dapat menggunakan hibah untuk memungkinkan prinsipal melihat kunci KMS, menggunakannya dalam operasi kriptografi, dan membuat dan menghentikan hibah. Untuk detailnya, lihat Operasi pemberian izin. Anda juga dapat menggunakan batasan hibah untuk membatasi izin dalam hibah untuk kunci enkripsi simetris.
- Perwakilan bisa mendapatkan izin untuk membuat pemberian izin dari kebijakan kunci atau kebijakan IAM. Kepala sekolah yang mendapatkan `kms:CreateGrant` izin dari kebijakan dapat membuat hibah untuk operasi hibah apa pun pada kunci KMS. Prinsipal ini tidak diharuskan memiliki izin yang mereka berikan pada kunci. Saat Anda memberikan izin `kms:CreateGrant` dalam kebijakan, Anda dapat menggunakan ketentuan kebijakan untuk membatasi izin ini.
- Perwakilan juga bisa mendapatkan izin untuk membuat pemberian izin dari sebuah pemberian izin. Perwakilan ini hanya dapat mendelegasikan izin yang diberikan, meskipun mereka memiliki izin lain dari kebijakan. Untuk detail selengkapnya, lihat Pemberian izin CreateGrant .

Untuk bantuan dengan konsep yang berkaitan dengan pemberian izin, lihat Terminologi pemberian izin.

Konsep hibah

Untuk menggunakan pemberian izin secara efektif, Anda harus memahami istilah dan konsep yang digunakan AWS KMS.

Batas pemberian izin

> Syarat yang membatasi izin dalam pemberian izin. Saat ini, AWS KMS mendukung batas pemberian izin berdasarkankonteks enkripsi dalam permintaan untuk operasi kriptografi. Untuk detail selengkapnya, lihat Menggunakan batas pemberian izin.

ID Pemberian izin

Pengidentifikasi unik hibah untuk kunci KMS. Anda dapat menggunakan ID hibah, bersama dengan pengenal kunci, untuk mengidentifikasi hibah dalam RevokeGrantpermintaan RetireGrantatau permintaan.

Operasi pemberian izin

Operasi AWS KMS yang dapat Anda izinkan dalam pemberian izin. Jika Anda menentukan operasi lain, CreateGrantpermintaan gagal dengan `ValidationError` pengecualian. Ini juga merupakan operasi yang menerima token izin. Untuk informasi selengkapnya tentang izin ini, lihat AWS KMS izin.

Operasi hibah ini sebenarnya mewakili izin untuk menggunakan operasi. Oleh karena itu, untuk operasi `ReEncrypt`, Anda dapat menentukan `ReEncryptFrom`, `ReEncryptTo`, atau kedua `ReEncrypt*`.

Operasi pemberian izin adalah:

- Operasi kriptografi
  - Dekripsi
  - Enkripsi
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyPairWithoutPlaintext
  - GenerateDataKeyWithoutPlaintext
  - GenerateMac
  - ReEncryptFrom
  - ReEncryptTo
  - Tanda
  - Verifikasi
  - VerifyMac
- Operasi lainnya
  - CreateGrant
  - DescribeKey
  - GetPublicKey
  - RetireGrant

Operasi hibah yang Anda izinkan harus didukung oleh kunci KMS dalam hibah. Jika Anda menentukan operasi yang tidak didukung, CreateGrantpermintaan gagal dengan `ValidationError` pengecualian. Misalnya, hibah untuk kunci KMS enkripsi simetris tidak dapat mengizinkan Tanda, Verifikasi, `GenerateMac`atau operasi. `VerifyMac` Hibah untuk kunci KMS asimetris tidak dapat mengizinkan operasi apa pun yang menghasilkan kunci data atau pasangan kunci data.

Token izin

AWS KMSAPI mengikuti model konsistensi akhirnya. Saat Anda membuat hibah, mungkin ada penundaan singkat sebelum perubahan tersedia di seluruhAWS KMS. Biasanya diperlukan waktu kurang dari beberapa detik agar perubahan menyebar ke seluruh sistem, tetapi dalam beberapa kasus dapat memakan waktu beberapa menit. Jika Anda mencoba menggunakan hibah sebelum sepenuhnya menyebar melalui sistem, Anda mungkin mendapatkan kesalahan akses ditolak. Token izin memungkinkan Anda mengacu pada pemberian izin dan segera menggunakan izin pemberian izin.

Token hibah adalah string unik, tidak rahasia, panjang variabel, berenkode base64 yang mewakili hibah. Anda dapat menggunakan token izin untuk mengidentifikasi pemberian izin dalam operasi pemberian izin. Namun, karena nilai token merupakan intisari hash, nilai tersebut tidak mengungkap detail tentang pemberian izin.

Token hibah dirancang untuk digunakan hanya sampai hibah telah sepenuhnya disebarkan. AWS KMS Setelah itu, perwakilan penerima dapat menggunakan izin dalam pemberian izin tanpa memberikan token izin atau bukti izin lain. Anda dapat menggunakan token izin kapan saja, tetapi setelah pemberian izin mencapai eventual consistency, AWS KMS akan menggunakan pemberian izin untuk menentukan izin, bukan token izin.

Misalnya, perintah berikut memanggil GenerateDataKeyoperasi. Ini menggunakan token hibah untuk mewakili hibah yang memberikan izin kepada penelepon (prinsipal penerima hibah) untuk memanggil `GenerateDataKey` kunci KMS yang ditentukan.

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

Anda juga dapat menggunakan token izin untuk mengidentifikasi pemberian izin dalam operasi yang mengelola pemberian izin. Misalnya, kepala sekolah yang pensiun dapat menggunakan token hibah dalam panggilan ke RetireGrantoperasi.

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

`CreateGrant` adalah satu-satunya operasi yang menampilkan token izin. Anda tidak bisa mendapatkan token hibah dari AWS KMS operasi lain atau dari peristiwa CloudTrail log untuk CreateGrant operasi. ListRetirableGrantsOperasi ListGrantsdan mengembalikan ID hibah, tetapi bukan token hibah.

Untuk detail selengkapnya, lihat Menggunakan token izin.

Perwakilan penerima

Identitas yang mendapatkan izin yang ditentukan dalam hibah. Setiap hibah memiliki satu pokok penerima hibah, tetapi pokok penerima hibah dapat mewakili banyak identitas.

Perwakilan penerima dapat berupa perwakilan AWS, termasuk Akun AWS (akar), Pengguna IAM, IAM Role, peran atau pengguna gabungan, atau pengguna peran diasumsikan. Pokok penerima hibah dapat berada di akun yang sama dengan kunci KMS atau akun yang berbeda. Namun, perwakilan penerima tidak boleh menjadi perwakilan layanan, grup IAM, atau AWSorganisasi.

> **Note**
>
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Penghentian (pemberian izin)

Mengakhiri pemberian izin. Anda menghentikan hibah ketika Anda selesai menggunakan izin.

Mencabut dan menghentikan pemberian izin akan menghapus pemberian izin tersebut. Namun penghentian dilakukan oleh perwakilan yang ditentukan dalam pemberian izin. Pencabutan biasanya dilakukan oleh administrator kunci. Untuk detail selengkapnya, lihat Menghentikan dan mencabut pemberian izin.

Perwakilan penghentian

Perwakilan yang dapat menghentikan pemberian izin. Anda dapat menentukan perwakilan penghentian dalam pemberian izin, tetapi tidak diperlukan. Prinsipal pensiun dapat berupa AWS

prinsipal apa pun, termasukAkun AWS, pengguna IAM, peran IAM, pengguna federasi, dan pengguna peran yang diasumsikan. Kepala sekolah yang pensiun dapat berada di akun yang sama dengan kunci KMS atau akun yang berbeda.

> **Note**
>
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Selain pensiun kepala sekolah yang ditentukan dalam hibah, hibah dapat dipensiunkan oleh Akun AWS di mana hibah itu dibuat. Jika pemberian izin mengizinkan operasi `RetireGrant`, perwakilan penerima dapat menghentikan pemberian izin. Juga, Akun AWS atau yang merupakan kepala sekolah Akun AWS yang pensiun dapat mendelegasikan izin untuk pensiun hibah kepada kepala sekolah IAM dalam hal yang sama. Akun AWS Untuk detail selengkapnya, lihat Menghentikan dan mencabut pemberian izin.

Mencabut (pemberian izin)

Mengakhiri pemberian izin. Anda mencabut pemberian izin untuk secara aktif menolak izin yang diberikan pemberian izin.

Mencabut dan menghentikan pemberian izin akan menghapus pemberian izin tersebut. Namun penghentian dilakukan oleh perwakilan yang ditentukan dalam pemberian izin. Pencabutan biasanya dilakukan oleh administrator kunci. Untuk detail selengkapnya, lihat Menghentikan dan mencabut pemberian izin.

Eventual consistency (untuk pemberian izin)

AWS KMSAPI mengikuti model konsistensi akhirnya. Saat Anda membuat, pensiun, atau mencabut hibah, mungkin ada penundaan singkat sebelum perubahan tersedia secara keseluruhan. AWS KMS Biasanya diperlukan waktu kurang dari beberapa detik agar perubahan menyebar ke seluruh sistem, tetapi dalam beberapa kasus dapat memakan waktu beberapa menit.

Anda mungkin menyadari penundaan singkat ini jika Anda mengalami kesalahan yang tidak terduga. Misalnya, jika Anda mencoba mengelola pemberian izin baru atau menggunakan izin dalam pemberian izin baru sebelum dikenal di seluruh AWS KMS, Anda akan mengalami

kesalahan akses ditolak. Jika Anda menghentikan atau mencabut pemberian izin, perwakilan penerima mungkin masih bisa menggunakan izin selama jangka waktu yang singkat sampai pemberian izin sepenuhnya dihapus. Strategi yang umum adalah untuk mencoba lagi permintaan, dan beberapa SDK AWS termasuk backoff otomatis dan logika pengulangan.

AWS KMS memiliki fitur yang dapat mengurangi penundaan singkat ini.

- Untuk segera menggunakan izin dalam pemberian izin baru, gunakan token izin. Anda dapat menggunakan token izin untuk mengacu pada pemberian izin dalam operasi pemberian izin. Untuk instruksi, lihat Menggunakan token izin.
- CreateGrantOperasi ini memiliki `Name` parameter yang mencegah operasi coba lagi membuat hibah duplikat.

> **Note**
>
> Token izin menggantikan validitas pemberian izin sampai semua titik akhir dalam layanan telah diperbarui dengan status izin baru. Dalam kebanyakan kasus, eventual consistency akan tercapai dalam waktu lima menit.

Untuk informasi lebih lanjut, lihat konsistensi AWS KMS akhirnya.

Praktik terbaik untuk AWS KMS hibah

AWS KMSmerekomendasikan praktik terbaik berikut saat membuat, menggunakan, dan mengelola hibah.

- Batasi izin dalam hibah kepada yang dibutuhkan oleh prinsipal penerima hibah. Gunakan prinsip akses yang paling tidak memiliki hak istimewa.
- Gunakan prinsipal penerima hibah tertentu, seperti peran IAM, dan berikan izin utama penerima hibah untuk hanya menggunakan operasi API yang mereka butuhkan.
- Gunakan batasan hibah konteks enkripsi untuk memastikan bahwa penelepon menggunakan kunci KMS untuk tujuan yang dimaksud. Untuk detail tentang cara menggunakan konteks enkripsi dalam permintaan untuk mengamankan data Anda, lihat Cara Melindungi Integritas Data Terenkripsi Anda dengan Menggunakan AWS Key Management Service dan EncryptionContext di Blog AWSKeamanan.

> **Tip**
>
> Gunakan batasan EncryptionContextEqualhibah bila memungkinkan. Kendala EncryptionContextSubsethibah lebih sulit digunakan dengan benar. Jika Anda perlu menggunakannya, baca dokumentasi dengan seksama dan uji batasan hibah untuk memastikannya berfungsi sebagaimana dimaksud.

- Hapus hibah duplikat. Hibah duplikat memiliki ARN kunci yang sama, tindakan API, prinsipal penerima hibah, konteks enkripsi, dan nama. Jika Anda pensiun atau mencabut hibah asli tetapi meninggalkan duplikat, sisa hibah duplikat merupakan eskalasi hak istimewa yang tidak diinginkan. Untuk menghindari duplikasi hibah saat mencoba kembali `CreateGrant` permintaan, gunakan parameter. `Name` Untuk mendeteksi hibah duplikat, gunakan operasi. ListGrants Jika Anda secara tidak sengaja membuat hibah duplikat, pensiun atau cabut sesegera mungkin.

> **Note**
>
> Hibah untuk kunci AWS terkelola mungkin terlihat seperti duplikat tetapi memiliki prinsip penerima hibah yang berbeda.
> Kolom `GranteePrincipal` dalam respons `ListGrants` biasanya berisi perwakilan penerima pemberian izin. Namun, saat perwakilan penerima dalam pemberian izin adalah layanan AWS, kolom `GranteePrincipal` berisi perwakilan layanan, yang mungkin mewakili beberapa perwakilan penerima yang berbeda.

- Ingatlah bahwa hibah tidak secara otomatis kedaluwarsa. Pensiun atau cabut hibah segera setelah izin tidak lagi diperlukan. Hibah yang tidak dihapus dapat menimbulkan risiko keamanan untuk sumber daya terenkripsi.

Membuat pemberian izin

Sebelum membuat pemberian izin, pelajari tentang opsi untuk menyesuaikan pemberian izin Anda. Anda dapat menggunakan batas pemberian izin untuk membatasi izin dalam pemberian izin. Selain itu, pelajari tentang pemberian izin `CreateGrant`. Perwakilan yang mendapatkan izin untuk membuat pemberian izin dari pemberian izin terbatas dalam pemberian izin yang dapat dibuat.

Topik

- Membuat pemberian izin

- Menggunakan batas pemberian izin
- Pemberian izin CreateGrant

Membuat pemberian izin

Untuk membuat hibah, hubungi CreateGrantoperasi. Tentukan kunci KMS, pokok penerima hibah, dan daftar operasi hibah yang diizinkan. Anda juga dapat menetapkan perwakilan penghentian opsional. Untuk menyesuaikan pemberian izin, gunakan parameter `Constraints` opsional untuk menentukan batas pemberian izin.

Saat Anda membuat, menghentikan, atau mencabut pemberian izin, akan ada penundaan singkat, biasanya kurang dari lima menit, sebelum perubahan tersedia di seluruh AWS KMS. Untuk informasi lebih lanjut, lihat Konsistensi akhir (untuk hibah).

Misalnya, `CreateGrant` perintah berikut membuat hibah yang memungkinkan pengguna yang berwenang untuk mengambil `keyUserRole` peran untuk memanggil operasi Dekripsi pada kunci KMS simetris yang ditentukan. Pemberian izin akan menggunakan parameter `RetiringPrincipal` untuk menunjuk perwakilan yang dapat menghentikan pemberian izin. Ini juga mencakup batas pemberian izin yang memberikan izin hanya saat konteks enkripsi dalam permintaan berisi `"Department": "IT"`.

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

Jika kode Anda mencoba lagi operasi `CreateGrant`, atau menggunakan SDK AWS yang secara otomatis mencoba lagi permintaan, gunakan parameter Nama opsional untuk mencegah agar duplikat pemberian izin tidak dibuat. Jika AWS KMS mendapat permintaan `CreateGrant` untuk pemberian izin dengan sifat yang sama sebagai pemberian izin yang ada, termasuk nama, mengenali permintaan sebagai pengulangan, dan tidak membuat pemberian izin baru. Anda tidak dapat menggunakan nilai `Name` untuk mengidentifikasi pemberian izin dalam setiap operasi AWS KMS.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam nama hibah. Ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

```
$ 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}
```

Untuk contoh kode yang mendemonstrasikan pembuatan tanda tangan dengan pemberian izin dalam beberapa bahasa pemrograman, lihat Bekerja dengan izin.

Menggunakan batas pemberian izin

Kendala hibah menetapkan ketentuan pada izin yang diberikan hibah kepada prinsipal penerima hibah. Batas pemberian izin menggantikan kunci syarat dalam kebijakan kunci atau kebijakan IAM. Setiap nilai batasan hibah dapat mencakup hingga 8 pasangan konteks enkripsi. Nilai konteks enkripsi di setiap batasan hibah tidak dapat melebihi 384 karakter.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

AWS KMSmendukung dua kendala hibah, `EncryptionContextEquals` dan`EncryptionContextSubset`, keduanya menetapkan persyaratan untuk konteks enkripsi dalam permintaan untuk operasi kriptografi.

Kendala hibah konteks enkripsi dirancang untuk digunakan dengan operasi hibah yang memiliki parameter konteks enkripsi.

- Kendala konteks enkripsi hanya valid dalam hibah untuk kunci KMS enkripsi simetris. Operasi kriptografi dengan kunci KMS lainnya tidak mendukung konteks enkripsi.

- Kendala konteks enkripsi diabaikan untuk `DescribeKey` dan `RetireGrant` operasi. `DescribeKey`dan `RetireGrant` tidak memiliki parameter konteks enkripsi, tetapi Anda dapat menyertakan operasi ini dalam hibah yang memiliki batasan konteks enkripsi.
- Anda dapat menggunakan kendala konteks enkripsi dalam hibah untuk operasi. `CreateGrant` Batasan konteks enkripsi mengharuskan setiap hibah yang dibuat dengan `CreateGrant` izin memiliki batasan konteks enkripsi yang sama ketat atau lebih ketat.

AWS KMSmendukung batasan hibah konteks enkripsi berikut.

EncryptionContextEquals

Gunakan `EncryptionContextEquals` untuk menentukan konteks enkripsi yang tepat untuk permintaan yang diizinkan.

`EncryptionContextEquals`mengharuskan pasangan konteks enkripsi dalam permintaan adalah kecocokan yang tepat dan peka huruf besar/kecil untuk pasangan konteks enkripsi dalam batasan hibah. Pasangan ini dapat muncul dalam urutan apa pun, tetapi kunci dan nilai dalam setiap pasangan tidak dapat bervariasi.

Misalnya, jika batasan `EncryptionContextEquals` hibah memerlukan pasangan konteks `"Department": "IT"` enkripsi, hibah mengizinkan permintaan dari jenis yang ditentukan hanya jika konteks enkripsi dalam permintaan persis. `"Department": "IT"`

EncryptionContextSubset

Gunakan `EncryptionContextSubset` untuk mewajibkan permintaan tersebut menyertakan pasangan konteks enkripsi tertentu.

`EncryptionContextSubset`mengharuskan permintaan tersebut menyertakan semua pasangan konteks enkripsi dalam batasan hibah (kecocokan yang tepat dan peka huruf besar/kecil), tetapi permintaan tersebut juga dapat memiliki pasangan konteks enkripsi tambahan. Pasangan ini dapat muncul dalam urutan apa pun, tetapi kunci dan nilai dalam setiap pasangan tidak dapat bervariasi.

Misalnya, jika batasan `EncryptionContextSubset` hibah memerlukan pasangan konteks `Department=IT` enkripsi, hibah mengizinkan permintaan dari jenis yang ditentukan saat konteks enkripsi dalam permintaan tersebut`"Department": "IT"`, atau disertakan `"Department": "IT"` bersama dengan pasangan konteks enkripsi lainnya, seperti. `"Department": "IT","Purpose": "Test"`

Untuk menentukan batasan konteks enkripsi dalam hibah untuk kunci KMS enkripsi simetris, gunakan `Constraints` parameter dalam operasi. CreateGrant Hibah yang dibuat perintah ini memberi pengguna yang berwenang untuk mengambil izin `keyUserRole` peran untuk memanggil operasi Dekripsi. Tetapi izin itu hanya efektif ketika konteks enkripsi dalam `Decrypt` permintaan adalah pasangan konteks `"Department": "IT"` enkripsi.

```
$ 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}
```

Pemberian izin yang dihasilkan akan terlihat seperti berikut. Perhatikan bahwa izin yang diberikan untuk `keyUserRole` peran hanya efektif jika `Decrypt` permintaan menggunakan pasangan konteks enkripsi yang sama yang ditentukan dalam batasan hibah. Untuk menemukan hibah pada kunci KMS, gunakan operasi. ListGrants

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

```
}
```

Untuk memenuhi batasan `EncryptionContextEquals` hibah, konteks enkripsi dalam permintaan `Decrypt` operasi harus `"Department": "IT"` berpasangan. Permintaan seperti berikut dari kepala sekolah penerima hibah akan memenuhi batasan `EncryptionContextEquals` hibah.

```
$ 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
```

Ketika batasan hibah`EncryptionContextSubset`, pasangan konteks enkripsi dalam permintaan harus menyertakan pasangan konteks enkripsi dalam batasan hibah, tetapi permintaan juga dapat menyertakan pasangan konteks enkripsi lainnya. Batasan hibah berikut mengharuskan salah satu pasangan konteks enkripsi dalam permintaan tersebut adalah. `"Deparment": "IT"`

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

Permintaan berikut dari kepala sekolah penerima hibah akan memenuhi kendala `EncryptionContextEqual` dan `EncryptionContextSubset` hibah dalam contoh ini.

```
$ 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
```

Namun, permintaan seperti berikut dari prinsipal penerima hibah akan memenuhi batasan `EncryptionContextSubset` hibah, tetapi itu akan gagal dalam kendala hibah. `EncryptionContextEquals`

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

```
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT,Purpose=Test
```

AWSlayanan sering menggunakan kendala konteks enkripsi dalam hibah yang memberi mereka izin untuk menggunakan kunci KMS di Anda. Akun AWS Misalnya, Amazon DynamoDB menggunakan hibah seperti berikut ini untuk mendapatkan izin menggunakan Kunci yang dikelola AWSfor DynamoDB di akun Anda. Batas pemberian izin `EncryptionContextSubset` dalam pemberian izin ini membuat izin dalam pemberian izin hanya efektif jika konteks enkripsi dalam permintaan menyertakan pasangan `"subscriberID": "111122223333"` dan `"tableName": "Services"`. Batasan hibah ini berarti bahwa hibah memungkinkan DynamoDB untuk menggunakan kunci KMS yang ditentukan hanya untuk tabel tertentu di tabel Anda. Akun AWS

Untuk mendapatkan output ini, jalankan ListGrantsoperasi pada Kunci yang dikelola AWS for DynamoDB di akun Anda.

```
$ 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"
        }
    ]
}
```

Pemberian izin CreateGrant

Pemberian izin dapat menyertakan izin untuk memanggil operasi `CreateGrant`. Namun saat perwakilan penerima mendapat izin untuk menelepon `CreateGrant` dari pemberian izin, bukan dari kebijakan, maka izin zintersebut dibatasi.

- Perwakilan penerima hanya dapat membuat pemberian izin yang mengizinkan beberapa atau semua operasi dalam pemberian izin induk.
- Batas pemberian izin dalam pemberian izin yang dibuat harus setidaknya seketat batas dalam pemberian izin induk.

Batasan ini tidak berlaku untuk perwakilan yang mendapatkan izin `CreateGrant` dari kebijakan, meskipun izinnya dapat dibatasi oleh ketentuan kebijakan.

Sebagai contoh, pertimbangkan pemberian izin yang memungkinkan perwakilan penerima memanggil operasi `GenerateDataKey`, `Decrypt`, dan `CreateGrant`. Kami memanggil pemberian izin yang memberi izin `CreateGrant` pada pemberian izin induk.

```
# 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"
                }
            },
        }
    ]
}
```

Perwakilan penerima, exampleUser, dapat menggunakan izin ini untuk membuat pemberian izin yang menyertakan setiap subset operasi yang ditentukan dalam pemberian izin asli, seperti `CreateGrant` dan `Decrypt`. Pemberian izin anaktidak dapat menyertakan operasi lain, seperti `ScheduleKeyDeletion` atau `ReEncrypt`.

Selain itu, batas pemberian izin dalam pemberian izin anak harus seketat atau lebih ketat dari batas dalam pemberian izin induk. Misalnya, pemberian izin anak dapat menambahkan pasangan ke batas `EncryptionContextSubset` dalam pemberian izin induk, tetapi tidak dapat menghapusnya. Pemberian izin anak dapat mengubah batas `EncryptionContextSubset` ke batas `EncryptionContextEquals`, tapi tidak sebaliknya.

Misalnya, perwakilan penerima dapat menggunakan izin `CreateGrant` yang didapat dari pemberian izin induk untuk membuat pemberian izin anak berikut. Operasi dalam pemberian izin anak adalah subset operasi dalam pemberian izin induk dan batas pemberian izin menjadi lebih ketat.

```
# 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"
                }
            },
        }
    ]
}
```

Perwakilan penerima dalam pemberian izin anak, `anotherUser`, dapat menggunakan izin `CreateGrant` untuk membuat pemberian izin. Namun, pemberian izin yang dibuat `anotherUser` harus menyertakan operasi dalam pemberian izin induknya atau subset, dan batas pemberian izin harus sama atau lebih ketat.

Mengelola pemberian izin

Perwakilan yang memiliki izin yang diperlukan dapat melihat, menggunakan, dan menghapus (menghentikan atau mencabut) pemberian izin. Untuk menyempurnakan izin pembuatan dan pengelolaan pemberian izin, AWS KMS mendukung beberapa ketentuan kebijakan yang dapat Anda gunakan dalam kebijakan kunci dan kebijakan IAM.

Topik

- Mengontrol akses ke pemberian izin
- Melihat pemberian izin
- Menggunakan token izin
- Menghentikan dan mencabut pemberian izin

Mengontrol akses ke pemberian izin

Anda dapat mengontrol akses ke operasi yang membuat dan mengelola pemberian izin dalam kebijakan kunci, kebijakan IAM, dan pemberian izin. Perwakilan yang mendapatkan izin `CreateGrant` dari pemberian izin memilikilebih banyak izin pemberian izin terbatas.

Operasi API	Kebijakan kunci atau kebijakan IAM	Pemberian Izin
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
Menghentikan Pemberian Izin	(Terbatas. LihatMenghentikan dan mencabut pemberian izin)	✓
RevokeGrant	✓	-

Saat menggunakan kebijakan kunci atau kebijakan IAM untuk mengontrol akses ke operasi yang membuat dan mengelola pemberian izin, Anda dapat menggunakan satu atau beberapa ketentuan kebijakan berikut untuk membatasi izin. AWS KMS mendukung semua kunci syarat terkait pemberian izin berikut. Untuk informasi dan contoh mendetail, lihat AWS KMS kunci kondisi.

km: GrantConstraintType

Memungkinkan perwakilan membuat pemberian izin hanya jika pemberian izin menyertakan batas pemberian izin yang ditentukan.

km: GrantIsFor AWSResource

Memungkinkan perwakilan memanggil `CreateGrant`, `ListGrants`, atau `RevokeGrant` hanya jika layanan AWS yang terintegrasi dengan AWS KMS mengirimkan permintaan atas nama perwakilan.

km: GrantOperations

Memungkinkan perwakilan membuat pemberian izin, tetapi membatasi pemberian izin untuk operasi yang ditentukan.

km: GranteePrincipal

Memungkinkan kepala sekolah membuat pemberian izin hanya untuk perwakilan penerima yang ditentukan.

km: RetiringPrincipal

Memungkinkan perwakilan membuat pemberian izin hanya jika pemberian izin menentukan perwakilan penghentian tertentu.

Melihat pemberian izin

Untuk melihat hibah, gunakan ListGrantsoperasi. Anda harus menentukan kunci KMS tempat hibah berlaku. Anda juga dapat memfilter daftar pemberian izin berdasarkan ID pemberian izin atau perwakilan penerima. Untuk contoh lainnya, lihat Melihat izin.

Untuk melihat semua hibah di Akun AWS dan Wilayah dengan kepala sekolah pensiun tertentu, gunakan. ListRetirableGrants Respons meliputi detail tentang setiap pemberian izin.

> **Note**
>
> Kolom `GranteePrincipal` dalam respons `ListGrants` biasanya berisi perwakilan penerima pemberian izin. Namun, saat perwakilan penerima dalam pemberian izin adalah layanan AWS, kolom `GranteePrincipal` berisi perwakilan layanan, yang mungkin mewakili beberapa perwakilan penerima yang berbeda.

Misalnya, perintah berikut mencantumkan semua hibah untuk kunci KMS.

```
$  aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
```

```
            "GranteePrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Operations": [
                "Decrypt"
            ]
        }
    ]
}
```

Menggunakan token izin

AWS KMSAPI mengikuti model konsistensi akhirnya. Saat Anda membuat pemberian izin, mungkin tidak langsung efektif. Mungkin ada penundaan singkat sebelum perubahan tersedia di seluruhAWS KMS. Biasanya diperlukan waktu kurang dari beberapa detik agar perubahan menyebar ke seluruh sistem, tetapi dalam beberapa kasus dapat memakan waktu beberapa menit. Setelah perubahan disebarkan sepenuhnya ke seluruh sistem, prinsipal penerima hibah dapat menggunakan izin dalam hibah tanpa menentukan token hibah atau bukti hibah apa pun. Namun, jika hibah yang sangat baru sehingga belum diketahui semua orangAWS KMS, permintaan tersebut mungkin gagal dengan `AccessDeniedException` kesalahan.

Untuk segera menggunakan izin dalam pemberian izin baru, gunakan token izin untuk pemberian izin. Simpan token hibah yang dikembalikan oleh CreateGrantoperasi. Lalu kirimkan token izin dalam permintaan untuk operasi AWS KMS. Anda dapat mengirimkan token izin keAWS KMS operasi pemberian izin dan Anda dapat mengirimkan beberapa token izin dalam permintaan yang sama.

Contoh berikut menggunakan `CreateGrant` operasi untuk membuat hibah yang memungkinkan GenerateDataKeydan Dekripsi operasi. Operasi tersebut menyimpan token izin yang menampilkan `CreateGrant` dalam variabel `token`. Kemudian, dalam panggilan ke operasi `GenerateDataKey`, ini menggunakan token izin di variabel `token`.

```
# Create a grant; save the grant token
$ token=$(aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/appUser \
    --retiring-principal arn:aws:iam::111122223333:user/acctAdmin \
    --operations GenerateDataKey Decrypt \
    --query GrantToken \
    --output text)

# Use the grant token in a request
$ aws kms generate-data-key \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --grant-tokens $token
```

Kepala sekolah dengan izin juga dapat menggunakan token hibah untuk pensiun hibah baru bahkan sebelum hibah tersedia. AWS KMS (Operasi `RevokeGrant` tidak menerima token izin.) Untuk detail selengkapnya, lihat Menghentikan dan mencabut pemberian izin.

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Menghentikan dan mencabut pemberian izin

Untuk menghapus, menghentikan, atau mencabut pemberian izin.

Operasi RetireGrantdan RevokeGrantoperasi sangat mirip satu sama lain. Kedua operasi menghapus hibah, yang menghilangkan izin yang diizinkan oleh hibah. Perbedaan utama antara operasi ini adalah bagaimana mereka diotorisasi.

RevokeGrant

Seperti kebanyakan operasi AWS KMS, akses ke operasi `RevokeGrant` dikontrol melalui kebijakan kunci dan Kebijakan IAM. RevokeGrantAPI dapat dipanggil oleh kepala sekolah mana pun dengan `kms:RevokeGrant` izin. Izin ini disertakan dalam izin standar yang diberikan kepada administrator kunci. Biasanya, administrator mencabut pemberian izin untuk menolak izin yang diberikan pemberian izin.

RetireGrant

Pemberian izin menentukan orang yang dapat menghentikannya. Desain ini memungkinkan Anda mengontrol siklus hidup pemberian izin tanpa mengubah kebijakan kunci atau kebijakan IAM. Biasanya, Anda akan menghentikan pemberian izin begitu selesai menggunakan izin.

Pemberian izin dapat dihentikan dengan perwakilan penghentian yang ditentukan dalam pemberian izin. Kepala penerima hibah juga dapat mempensiunkan hibah, tetapi hanya jika mereka juga merupakan kepala sekolah pensiun atau hibah termasuk operasi. `RetireGrant` Sebagai cadangan, Akun AWS di mana hibah dibuat dapat menghentikan hibah.

Ada izin `kms:RetireGrant` yang dapat digunakan dalam kebijakan IAM, tetapi memiliki utilitas terbatas. Perwakilan yang ditentukan dalam pemberian izin dapat menghentikan pemberian izin tanpa izin `kms:RetireGrant`. Izin `kms:RetireGrant` saja tidak memungkinkan kepala sekolah

untuk menghentikan pemberian izin. Izin `kms:RetireGrant` tidak akan efektif dalam kebijakan kunci.

- Untuk menolak izin penghentian pemberian izin, Anda dapat menggunakan tindakan `Deny` dengan izin `kms:RetireGrant`.
- Akun AWSYang memiliki kunci KMS dapat mendelegasikan `kms:RetireGrant` izin kepada kepala IAM di akun.
- Jika kepala sekolah pensiun berbedaAkun AWS, administrator di akun lain dapat menggunakan `kms:RetireGrant` untuk mendelegasikan izin untuk mempensiunkan hibah ke kepala IAM di akun itu.

AWS KMSAPI mengikuti model konsistensi akhirnya. Saat Anda membuat, pensiun, atau mencabut hibah, mungkin ada penundaan singkat sebelum perubahan tersedia secara keseluruhan. AWS KMS Biasanya diperlukan waktu kurang dari beberapa detik agar perubahan menyebar ke seluruh sistem, tetapi dalam beberapa kasus dapat memakan waktu beberapa menit. Jika Anda perlu segera menghapus hibah baru, sebelum tersedia secara keseluruhanAWS KMS, gunakan token hibah untuk menghentikan hibah. Anda tidak dapat menggunakan token izin untuk mencabut pemberian izin.

Terhubung ke AWS KMS melalui VPC endpoint

Anda dapat terhubung langsung AWS KMS melalui titik akhir antarmuka pribadi di cloud pribadi virtual (VPC) Anda. Bila Anda menggunakan antarmuka VPC endpoint, komunikasi antara VPC Anda dan AWS KMS dilakukan sepenuhnya dalam jaringan. AWS

AWS KMSmendukung titik akhir Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Masing-masing VPC endpoint diwakili oleh satu atau lebih Antarmuka Jaringan Elastis (ENI) dengan alamat IP privat di subnet VPC Anda.

Titik akhir VPC antarmuka menghubungkan VPC Anda secara langsung AWS KMS tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan AWS KMS.

Wilayah

AWS KMSmendukung kebijakan titik akhir VPC dan titik akhir VPC di semua yang didukung. Wilayah AWS AWS KMS

Topik

- Pertimbangan untuk VPC endpoint AWS KMS

- Membuat VPC endpoint untuk AWS KMS
- Terhubung ke VPC endpoint AWS KMS
- Mengontrol akses ke VPC endpoint
- Menggunakan VPC endpoint dalam pernyataan kebijakan
- Mencatat VPC endpoint Anda

Pertimbangan untuk VPC endpoint AWS KMS

Sebelum Anda menyiapkan titik akhir VPC antarmuka untukAWS KMS, tinjau topik properti dan batasan titik akhir Antarmuka di Panduan. AWS PrivateLink

AWS KMSdukungan untuk titik akhir VPC mencakup yang berikut ini.

- Anda dapat menggunakan titik akhir VPC untuk memanggil semua operasi AWS KMS API dari VPC Anda.
- Anda dapat membuat titik akhir VPC antarmuka yang terhubung ke titik akhir AWS KMS wilayah atau titik akhir FIPS. AWS KMS
- Anda dapat menggunakan AWS CloudTrail log untuk mengaudit penggunaan kunci KMS melalui titik akhir VPC. Untuk rincian selengkapnya, lihat Mencatat VPC endpoint Anda.

Membuat VPC endpoint untuk AWS KMS

Anda dapat membuat VPC endpoint untuk AWS KMS menggunakan konsol Amazon VPC atau API Amazon VPC. Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLinkPanduan.

- Untuk membuat VPC endpoint untuk AWS KMS, gunakan nama layanan berikut:

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

  Sebagai contoh, di Wilayah US West (Oregon) (`us-west-2`), nama layanan akan menjadi:

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

- Untuk membuat titik akhir VPC yang terhubung ke titik akhir AWS KMSFIPS, gunakan nama layanan berikut:

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

Sebagai contoh, di Wilayah US West (Oregon) (`us-west-2`), nama layanan akan menjadi:

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

Untuk mempermudah penggunaan titik akhir VPC, Anda dapat mengaktifkan nama DNS pribadi untuk titik akhir VPC Anda. Jika Anda memilih opsi Aktifkan Nama DNS, nama host AWS KMS DNS standar akan diselesaikan ke titik akhir VPC Anda. Misalnya, `https://kms.us-west-2.amazonaws.com` akan menyelesaikan ke titik akhir VPC yang terhubung ke nama layanan. `com.amazonaws.us-west-2.kms`

Opsi ini mempermudah untuk menggunakan VPC endpoint. SDK AWS dan AWS CLI menggunakan nama host DNS AWS KMS standar secara default, sehingga Anda tidak perlu menentukan URL VPC endpoint dalam aplikasi dan perintah.

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka di Panduan. AWS PrivateLink

Terhubung ke VPC endpoint AWS KMS

Anda dapat terhubung ke AWS KMS melalui VPC endpoint dengan menggunakan SDK AWS, AWS CLI atau AWS Tools for PowerShell. Untuk menentukan VPC endpoint, gunakan nama DNS-nya.

Misalnya, perintah kunci-daftar ini menggunakan parameter `endpoint-url` untuk menentukan VPC endpoint. Untuk menggunakan perintah seperti ini, ganti contoh ID VPC endpoint dengan yang ada di akun Anda.

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

Jika Anda mengaktifkan nama host privat ketika Anda membuat VPC endpoint Anda, Anda tidak perlu menentukan URL VPC endpoint di perintah CLI atau konfigurasi aplikasi. Nama host AWS KMS DNS standar diselesaikan ke titik akhir VPC Anda. SDK AWS CLI dan SDK menggunakan nama host ini secara default, sehingga Anda dapat mulai menggunakan titik akhir VPC untuk terhubung ke titik akhir AWS KMS regional tanpa mengubah apa pun di skrip dan aplikasi Anda.

Untuk menggunakan nama host pribadi, `enableDnsSupport` atribut `enableDnsHostnames` dan VPC Anda harus disetel ke. `true` Untuk mengatur atribut ini, gunakan ModifyVpcAttributeoperasi. Untuk detailnya, lihat Melihat dan memperbarui atribut DNS untuk VPC Anda di Panduan Pengguna Amazon VPC.

Mengontrol akses ke VPC endpoint

Untuk mengontrol akses ke VPC endpoint Anda untuk AWS KMS, lampirkan Kebijakan VPC endpoint ke VPC endpoint. Kebijakan titik akhir menentukan apakah prinsipal dapat menggunakan VPC endpoint untuk memanggil operasi AWS KMS pada sumber daya AWS KMS.

Anda dapat membuat kebijakan VPC endpoint ketika Anda membuat titik akhir Anda, dan Anda dapat mengubah kebijakan VPC endpoint setiap saat. Gunakan konsol manajemen VPC, atau operasi atau. CreateVpcEndpointModifyVpcEndpoint Anda juga dapat membuat dan mengubah kebijakan VPC endpoint dengan menggunakan templat AWS CloudFormation. Untuk bantuan menggunakan konsol manajemen VPC, lihat Membuat titik akhir antarmuka dan Memodifikasi titik akhir antarmuka dalam Panduan. AWS PrivateLink

> **Note**
>
> AWS KMS mendukung kebijakan VPC endpoint dimulai pada bulan Juli 2020. VPC endpoint untuk AWS KMS yang dibuat sebelum tanggal tersebut memiliki Kebijakan VPC endpoint default, tetapi Anda bisa mengubahnya kapan saja.

Untuk mendapatkan bantuan mengenai cara menulis dan memformat dokumen kebijakan JSON, lihat Referensi Kebijakan IAM JSON dalam Panduan Pengguna IAM.

Topik

- Tentang kebijakan VPC endpoint
- Kebijakan VPC endpoint default
- Membuat kebijakan VPC endpoint
- Melihat kebijakan VPC endpoint

Tentang kebijakan VPC endpoint

Untuk permintaan AWS KMS yang menggunakan VPC endpoint agar berhasil, prinsipal memerlukan izin dari dua sumber:

- Kebijakan kunci, kebijakan IAM, atau hibah harus memberikan izin utama untuk memanggil operasi pada sumber daya (kunci KMS atau alias).
- Kebijakan VPC endpoint harus memberikan prinsipal izin untuk menggunakan titik akhir untuk membuat permintaan.

Misalnya, kebijakan kunci mungkin memberikan izin utama untuk memanggil Dekripsi pada kunci KMS tertentu. Namun, kebijakan titik akhir VPC mungkin tidak mengizinkan prinsipal tersebut untuk memanggil `Decrypt` kunci KMS tersebut dengan menggunakan titik akhir.

Atau kebijakan titik akhir VPC mungkin mengizinkan prinsipal menggunakan titik akhir untuk memanggil DisableKeykunci KMS tertentu. Tetapi jika prinsipal tidak memiliki izin tersebut dari kebijakan kunci, kebijakan IAM, atau hibah, permintaan gagal.

Kebijakan VPC endpoint default

Setiap VPC endpoint memiliki kebijakan VPC endpoint, tetapi Anda tidak diharuskan untuk menentukan kebijakan. Jika Anda tidak menentukan kebijakan, kebijakan titik akhir default memungkinkan semua operasi oleh semua prinsipal di semua sumber daya pada titik akhir.

Namun, untuk sumber daya AWS KMS, prinsipal juga harus memiliki izin untuk memanggil operasi dari kebijakan utama, Kebijakan IAM, atau hibah. Oleh karena itu, dalam praktik, kebijakan default mengatakan bahwa jika prinsipal memiliki izin untuk memanggil operasi pada sumber daya, mereka juga dapat memanggilnya dengan menggunakan titik akhir.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Untuk mengizinkan prinsipal menggunakan titik akhir VPC hanya untuk sebagian dari operasi yang diizinkan, buat atau perbarui kebijakan titik akhir VPC.

Membuat kebijakan VPC endpoint

Kebijakan VPC endpoint menentukan apakah prinsipal memiliki izin untuk menggunakan VPC endpoint untuk melakukan operasi pada sumber daya. Untuk sumber daya AWS KMS, prinsipal juga harus memiliki izin untuk melakukan operasi dari kebijakan utama, Kebijakan IAM, atau hibah.

Setiap pernyataan kebijakan VPC endpoint memerlukan unsur-unsur berikut:

- Prinsip-prinsip yang dapat melakukan tindakan
- Tindakan yang dapat dilakukan
- Sumber daya yang dapat digunakan untuk mengambil tindakan

Pernyataan kebijakan tidak menentukan VPC endpoint. Sebaliknya, berlaku untuk VPC endpoint di mana kebijakan tersebut terpasang. Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Berikut adalah contoh kebijakan VPC endpoint untuk AWS KMS. Saat dilampirkan ke titik akhir VPC, kebijakan ini memungkinkan `ExampleUser` untuk menggunakan titik akhir VPC untuk memanggil operasi yang ditentukan pada kunci KMS yang ditentukan. Sebelum menggunakan kebijakan seperti ini, ganti contoh prinsipal dan ARN kunci dengan nilai yang valid dari akun Anda.

```
{
   "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 mencatat semua operasi yang menggunakan VPC endpoint. Namun, CloudTrail log Anda tidak menyertakan operasi yang diminta oleh kepala sekolah di akun lain atau operasi untuk kunci KMS di akun lain.

Dengan demikian, Anda mungkin ingin membuat kebijakan VPC endpoint yang mencegah prinsipal di akun eksternal menggunakan VPC endpoint untuk memanggil operasi AWS KMS apa pun pada kunci apa pun di akun lokal.

Contoh berikut menggunakan aws: PrincipalAccount global condition key untuk menolak akses ke semua prinsipal untuk semua operasi pada semua kunci KMS kecuali prinsipal ada di akun lokal. Sebelum menggunakan kebijakan seperti ini, ganti ID akun contoh dengan yang valid.

```
{
  "Statement": [
    {
      "Sid": "AccessForASpecificAccount",
      "Principal": {"AWS": "*"},
      "Action": "kms:*",
      "Effect": "Deny",
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "111122223333"
        }
      }
    }
  ]
}
```

Melihat kebijakan VPC endpoint

Untuk melihat kebijakan titik akhir VPC untuk titik akhir, gunakan konsol manajemen VPC atau pengoperasiannya. DescribeVpcEndpoints

Perintah AWS CLI berikut ini mendapat kebijakan untuk titik akhir dengan ID VPC endpoint yang ditentukan.

Sebelum menggunakan perintah ini, ganti ID titik akhir contoh dengan yang valid dari akun Anda.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

Menggunakan VPC endpoint dalam pernyataan kebijakan

Anda dapat mengontrol akses ke sumber daya dan operasi AWS KMS ketika permintaan berasal dari VPC atau menggunakan VPC endpoint. Untuk melakukannya, gunakan salah satu kunci kondisi global berikut dalam kebijakan kunci atau kebijakan IAM.

- Gunakan kunci kondisi `aws:sourceVpce` untuk memberikan atau membatasi akses berdasarkan VPC endpoint.
- Gunakan kunci kondisi `aws:sourceVpc` untuk memberikan atau membatasi akses berdasarkan VPC yang menjadi host endpoint privat.

> **Note**
>
> Berhati-hatilah saat membuat kebijakan kunci dan kebijakan IAM berdasarkan VPC endpoint Anda. Jika pernyataan kebijakan mengharuskan permintaan berasal dari VPC atau VPC endpoint tertentu, permintaan dari layanan AWS terintegrasi yang menggunakan sumber daya AWS KMS atas nama Anda mungkin gagal. Untuk bantuan, lihat Menggunakan syarat VPC endpoint dalam kebijakan dengan izin AWS KMS.
> Selain itu, kunci syarat `aws:sourceIP` tidak efektif bila permintaan berasal dari Amazon VPC endpoint. Untuk membatasi permintaan ke VPC endpoint, gunakan kunci kondisi `aws:sourceVpce` atau `aws:sourceVpc`. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk titik akhir VPC dan layanan titik akhir VPC di Panduan. AWS PrivateLink

Anda dapat menggunakan kunci kondisi global ini untuk mengontrol akses ke AWS KMS keys (kunci KMS), alias, dan operasi seperti CreateKeyitu tidak bergantung pada sumber daya tertentu.

Misalnya, kebijakan kunci sampel berikut memungkinkan pengguna untuk melakukan beberapa operasi kriptografi dengan kunci KMS hanya ketika permintaan menggunakan titik akhir VPC yang ditentukan. Ketika pengguna membuat permintaan ke AWS KMS, ID VPC endpoint dalam permintaan dibandingkan dengan nilai kunci kondisi `aws:sourceVpce` dalam kebijakan. Jika tidak cocok, permintaan ditolak.

Untuk menggunakan kebijakan seperti ini, ganti placeholder ID Akun AWS dan ID VPC endpoint dengan nilai yang valid untuk akun Anda.

```
{
```

```
    "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"
                }
            }
        }

    ]
}
```

Anda juga dapat menggunakan tombol `aws:sourceVpc` kondisi untuk membatasi akses ke kunci KMS Anda berdasarkan VPC tempat titik akhir VPC berada.

Kebijakan kunci sampel berikut memungkinkan perintah yang mengelola kunci KMS hanya ketika mereka berasal`vpc-12345678`. Selain itu, ini memungkinkan perintah yang menggunakan kunci KMS untuk operasi kriptografi hanya ketika mereka berasal. `vpc-2b2b2b2b` Anda mungkin menggunakan kebijakan seperti ini jika aplikasi berjalan dalam satu VPC, tetapi Anda menggunakan VPC terisolasi kedua untuk fungsi manajemen.

Untuk menggunakan kebijakan seperti ini, ganti placeholder ID Akun AWS dan ID VPC endpoint dengan nilai yang valid untuk akun Anda.

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

```
    ]
}
```

Mencatat VPC endpoint Anda

AWS CloudTrail mencatat semua operasi yang menggunakan VPC endpoint. Ketika permintaan ke AWS KMS menggunakan VPC endpoint, ID VPC endpoint muncul di entri log AWS CloudTrail yang mencatat permintaan. Anda dapat menggunakan ID titik akhir untuk mengaudit penggunaan VPC endpoint AWS KMS.

Namun, CloudTrail log Anda tidak menyertakan operasi yang diminta oleh prinsipal di akun lain atau permintaan AWS KMS operasi pada kunci KMS dan alias di akun lain. Juga, untuk melindungi VPC Anda, permintaan yang ditolak oleh kebijakan VPC endpoint, tetapi sebaliknya akan diizinkan, tidak dicatat dalam AWS CloudTrail.

Misalnya, entri log contoh ini mencatat GenerateDataKeypermintaan yang menggunakan titik akhir VPC. Bidang `vpcEndpointId` muncul di akhir entri log.

```
{
  "eventVersion":"1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime":"2018-01-16T05:46:57Z",
  "eventSource":"kms.amazonaws.com",
  "eventName":"GenerateDataKey",
  "awsRegion":"eu-west-1",
  "sourceIPAddress":"172.01.01.001",
  "userAgent":"aws-cli/1.14.23 Python/2.7.12 Linux/4.9.75-25.55.amzn1.x86_64
 botocore/1.8.27",
  "requestParameters":{
    "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab",
    "numberOfBytes":128
  },
  "responseElements":null,
  "requestID":"a9fff0bf-fa80-11e7-a13c-afcabff2f04c",
  "eventID":"77274901-88bc-4e3f-9bb6-acf1c16f6a7c",
```

```
  "readOnly":true,
  "resources":[{
    "ARN":"arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId":"111122223333",
    "type":"AWS::KMS::Key"
  }],
  "eventType":"AwsApiCall",
  "recipientAccountId":"111122223333",
  "vpcEndpointId": "vpce-1234abcdf5678c90a"
}
```

Kunci kondisi untuk AWS KMS

Anda dapat menentukan kondisi dalam kebijakan utama dan kebijakan IAM yang mengontrol akses ke AWS KMS sumber daya. Pernyataan kebijakan hanya efektif jika syaratnya benar. Misalnya, Anda mungkin ingin pernyataan kebijakan berlaku hanya setelah tanggal tertentu. Atau, Anda mungkin ingin pernyataan kebijakan mengontrol akses hanya saat nilai tertentu muncul dalam permintaan API.

Untuk menentukan kondisi, Anda menggunakan kunci kondisi dalam `Condition`elemen pernyataan kebijakan dengan operator kondisi IAM. Beberapa kunci kondisi berlaku umumnya untuk AWS; yang lain khusus untuk AWS KMS.

Nilai kunci kondisi harus mematuhi karakter dan aturan pengkodean untuk kebijakan AWS KMS utama dan kebijakan IAM. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama. Untuk detail tentang aturan dokumen kebijakan IAM, lihat Persyaratan nama IAM di Panduan Pengguna IAM..

Topik

- AWS kunci kondisi global
- AWS KMS kunci kondisi
- AWS KMS kunci kondisi untuk AWS Nitro Enclave

AWS kunci kondisi global

AWS mendefinisikan kunci kondisi global, satu set kunci kondisi kebijakan untuk semua AWS layanan yang menggunakan IAM untuk kontrol akses. AWS KMS mendukung semua kunci kondisi global. Anda dapat menggunakannya dalam kebijakan AWS KMS utama dan kebijakan IAM.

Misalnya, Anda dapat menggunakan kunci kondisi PrincipalArn global aws: untuk mengizinkan akses ke AWS KMS key (kunci KMS) hanya jika prinsipal dalam permintaan diwakili oleh Amazon Resource Name (ARN) dalam nilai kunci kondisi. Untuk mendukung kontrol akses berbasis atribut (ABAC) di AWS KMS, Anda dapat menggunakan kunci kondisi global aws:ResourceTag/tag-key dalam kebijakan IAM untuk mengizinkan akses ke kunci KMS dengan tag tertentu.

Untuk membantu mencegah AWS layanan digunakan sebagai wakil yang bingung dalam kebijakan di mana prinsipal adalah kepala AWS layanan, Anda dapat menggunakan aws:SourceArnatau kunci kondisi aws:SourceAccountglobal. Lihat perinciannya di Menggunakan `aws:SourceArn` atau `aws:SourceAccount` mengkondisikan kunci.

Untuk informasi tentang kunci kondisi AWS global, termasuk jenis permintaan yang tersedia, lihat Kunci Konteks Kondisi AWS Global di Panduan Pengguna IAM. Untuk contoh penggunaan kunci syarat global dalam kebijakan IAM, lihat Mengontrol Akses ke Permintaan dan Mengontrol Kunci Tag di Panduan Pengguna IAM.

Topik berikut memberikan panduan khusus untuk menggunakan kunci syarat berdasarkan alamat IP dan VPC endpoint.

Topik

- Menggunakan syarat alamat IP dalam kebijakan dengan izin AWS KMS
- Menggunakan syarat VPC endpoint dalam kebijakan dengan izin AWS KMS

Menggunakan syarat alamat IP dalam kebijakan dengan izin AWS KMS

Anda dapat menggunakan AWS KMS untuk melindungi data Anda dalam AWS layanan terintegrasi. Tetapi berhati-hatilah saat menentukan operator kondisi alamat IP atau kunci `aws:SourceIp` kondisi dalam pernyataan kebijakan yang sama yang memungkinkan atau menolak akses ke. AWS KMS Misalnya, kebijakan di AWS: Denies Access to AWS Based on the Source IP membatasi AWS tindakan ke permintaan dari rentang IP yang ditentukan.

Pertimbangkan skenario ini:

1. Anda melampirkan kebijakan seperti yang ditampilkan di AWS: Denies Access to AWS Based on the Source IP ke identitas IAM. Anda menetapkan nilai kunci syarat `aws:SourceIp` ke rentang alamat IP untuk perusahaan pengguna. Identitas IAM ini memiliki kebijakan lain yang dilampirkan yang memungkinkannya menggunakan Amazon EBS, Amazon EC2, dan. AWS KMS

2. Identitas mencoba melampirkan volume EBS terenkripsi ke instans EC2. Tindakan ini gagal dengan kesalahan otorisasi meskipun pengguna memiliki izin untuk menggunakan semua layanan yang relevan.

Langkah 2 gagal karena permintaan AWS KMS untuk mendekripsi kunci data terenkripsi volume berasal dari alamat IP yang terkait dengan infrastruktur Amazon EC2. Agar berhasil, permintaan harus berasal dari alamat IP pengguna asal. Karena kebijakan di langkah 1 secara eksplisit menolak semua permintaan dari alamat IP selain yang ditentukan, izin Amazon EC2 untuk mendekripsi kunci data terenkripsi volume EBS ditolak.

Selain itu, kunci syarat `aws:sourceIP` tidak efektif bila permintaan berasal dari Amazon VPC endpoint. Untuk membatasi permintaan ke VPC endpoint, termasuk AWS KMS VPC endpoint, gunakan kunci syarat `aws:sourceVpce` atau `aws:sourceVpc`. Untuk informasi selengkapnya, lihat VPC endpoint - Mengontrol Penggunaan Titik Akhir di Panduan Pengguna Amazon VPC.

Menggunakan syarat VPC endpoint dalam kebijakan dengan izin AWS KMS

AWS KMS mendukung titik akhir Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Anda dapat menggunakan kunci kondisi global berikut dalam kebijakan utama dan kebijakan IAM untuk mengontrol akses ke AWS KMS sumber daya saat permintaan berasal dari VPC atau menggunakan titik akhir VPC. Untuk detail selengkapnya, lihat Menggunakan VPC endpoint dalam pernyataan kebijakan.

- `aws:SourceVpc` membatasi akses ke permintaan dari VPC yang ditentukan.
- `aws:SourceVpce` membatasi akses ke permintaan dari VPC endpoint yang ditentukan.

Jika Anda menggunakan tombol kondisi ini untuk mengontrol akses ke kunci KMS, Anda mungkin secara tidak sengaja menolak akses ke AWS layanan yang digunakan AWS KMS atas nama Anda.

Berhati-hatilah untuk menghindari situasi seperti contoh kunci syarat alamat IP. Jika Anda membatasi permintaan kunci KMS ke titik akhir VPC atau VPC, panggilan AWS KMS ke dari layanan terintegrasi, seperti Amazon S3 atau Amazon EBS, mungkin gagal. Hal ini dapat terjadi meskipun permintaan sumber pada akhirnya berasal dari VPC atau dari VPC endpoint.

AWS KMS kunci kondisi

AWS KMS menyediakan satu set kunci kondisi yang dapat Anda gunakan dalam kebijakan utama dan kebijakan IAM. Kunci kondisi ini khusus untuk AWS KMS. Misalnya, Anda dapat menggunakan

kunci `kms:EncryptionContext:`*`context-key`* kondisi untuk memerlukan konteks enkripsi tertentu saat mengontrol akses ke kunci KMS enkripsi simetris.

Ketentuan untuk permintaan operasi API

Banyak tombol AWS KMS kondisi mengontrol akses ke kunci KMS berdasarkan nilai parameter dalam permintaan AWS KMS operasi. Misalnya, Anda dapat menggunakan kunci KeySpec kondisi kms: dalam kebijakan IAM untuk mengizinkan penggunaan CreateKeyoperasi hanya jika nilai `KeySpec` parameter dalam `CreateKey` permintaan adalah. `RSA_4096`

Jenis syarat ini berfungsi bahkan saat parameter tidak muncul dalam permintaan, seperti saat Anda menggunakan nilai default parameter. Misalnya Anda dapat menggunakan kms: KeySpec condition key untuk memungkinkan pengguna menggunakan `CreateKey` operasi hanya jika nilai `KeySpec` parameternya`SYMMETRIC_DEFAULT`, yang merupakan nilai default. Syarat ini memungkinkan permintaan yang memiliki parameter `KeySpec` dengan nilai `SYMMETRIC_DEFAULT` dan permintaan yang tidak memiliki parameter `KeySpec`.

Ketentuan untuk kunci KMS yang digunakan dalam operasi API

Beberapa tombol AWS KMS kondisi dapat mengontrol akses ke operasi berdasarkan properti kunci KMS yang digunakan dalam operasi. Misalnya, Anda dapat menggunakan KeyOrigin kondisi kms: untuk mengizinkan prinsipal memanggil GenerateDataKeykunci KMS hanya ketika kunci KMS berada`Origin`. `AWS_KMS` Untuk mengetahui apakah kunci syarat dapat digunakan dengan cara ini, lihat deskripsi kunci syarat.

Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi. Jika Anda menggunakan jenis kunci kondisi ini dengan operasi yang tidak diizinkan untuk sumber daya kunci KMS tertentu, seperti ListKeys, izin tidak efektif karena kondisi tidak pernah dapat dipenuhi. Tidak ada sumber daya kunci KMS yang terlibat dalam otorisasi `ListKeys` operasi dan tidak ada `KeySpec` properti.

Topik berikut menjelaskan setiap kunci AWS KMS kondisi dan menyertakan contoh pernyataan kebijakan yang menunjukkan sintaks kebijakan.

Menggunakan operator set dengan kunci syarat

Jika kondisi kebijakan membandingkan dua set nilai, seperti kumpulan tag dalam permintaan dan kumpulan tag dalam kebijakan, Anda perlu memberi tahu AWS cara membandingkan set. IAM mendefinisikan dua operator set, `ForAnyValue` dan `ForAllValues`, untuk tujuan ini. Gunakan

operator set hanya dengan kunci syarat multi-nilai, yang membutuhkannya. Jangan gunakan operator set dengan kunci syarat bernilai tunggal. Seperti biasa, uji pernyataan kebijakan Anda secara menyeluruh sebelum menggunakannya dalam lingkungan produksi.

Kunci syarat adalah bernilai tunggal atau multi-nilai. Untuk menentukan apakah kunci AWS KMS kondisi bernilai tunggal atau multi-nilai, lihat kolom Jenis nilai dalam deskripsi kunci kondisi.

- Kunci syarat bernilai tunggal memiliki paling banyak satu nilai dalam konteks otorisasi (permintaan atau sumber daya). Misalnya, karena setiap panggilan API hanya dapat berasal dari satu Akun AWS, kms: CallerAccount adalah kunci kondisi bernilai tunggal. Jangan gunakan operator set dengan kunci syarat bernilai tunggal.
- Kunci syarat bernilai tunggal memiliki paling banyak satu nilai dalam konteks otorisasi (permintaan atau sumber daya). Misalnya, karena setiap kunci KMS dapat memiliki beberapa alias, kms: ResourceAliases dapat memiliki beberapa nilai. Kunci syarat multi-nilai memerlukan operator set.

Perhatikan bahwa perbedaan antara kunci syarat bernilai tunggal dan multi-nilai bergantung pada jumlah nilai dalam konteks otorisasi; bukan jumlah nilai dalam syarat kebijakan.

> **Warning**
>
> Menggunakan operator set dengan kunci syarat bernilai tunggal dapat membuat pernyataan kebijakan yang terlalu permisif (atau terlalu ketat). Gunakan operator set hanya dengan kunci syarat multi-nilai.
> Jika Anda membuat atau memperbarui kebijakan yang menyertakan operator `ForAllValues` set dengan kms:EncryptionContext: context-key atau `aws:RequestTag/`*`tag-key`* condition keys, AWS KMS menampilkan pesan galat berikut: `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.`

Untuk detail informasi tentang operator set `ForAnyValue` dan `ForAllValues`, lihat Menggunakan beberapa kunci dan nilai di Panduan Pengguna IAM. Untuk informasi tentang risiko menggunakan operator yang `ForAllValues` ditetapkan dengan kondisi bernilai tunggal, lihat Peringatan Keamanan — ForAllValues dengan kunci bernilai tunggal dalam Panduan Pengguna IAM.

Topik

- km: BypassPolicyLockoutSafetyCheck
- km: CallerAccount
- kms: CustomerMasterKeySpec (usang)
- kms: CustomerMasterKeyUsage (usang)
- km: DataKeyPairSpec
- km: EncryptionAlgorithm
- kms:EncryptionContext: kunci-konteks
- km: EncryptionContextKeys
- km: ExpirationModel
- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: GranteePrincipal
- km: KeyOrigin
- km: KeySpec
- km: KeyUsage
- 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 kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:BypassPolicyLockoutSafetyCheck`	Boolean	Bernilai tunggal	`CreateKey` `PutKeyPolicy`	Kebijakan IAM saja Kebijakan kunci dan kebijakan IAM

Kunci `kms:BypassPolicyLockoutSafetyCheck` kondisi mengontrol akses ke CreateKeydan PutKeyPolicyoperasi berdasarkan nilai `BypassPolicyLockoutSafetyCheck` parameter dalam permintaan.

Contoh berikut pernyataan kebijakan IAM mencegah pengguna melewati pemeriksaan keamanan penguncian kebijakan dengan menolak izin mereka untuk membuat kunci KMS ketika nilai parameter dalam permintaan adalah `BypassPolicyLockoutSafetyCheck` `CreateKey` `true`.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
```

```
}
```

Anda juga dapat menggunakan kunci syarat `kms:BypassPolicyLockoutSafetyCheck` dalam kebijakan IAM atau kebijakan kunci untuk mengontrol akses ke operasi `PutKeyPolicy`. Contoh pernyataan kebijakan berikut dari kebijakan utama mencegah pengguna melewati pemeriksaan keamanan penguncian kebijakan saat mengubah kebijakan kunci KMS.

Sebagai ganti dari menggunakan `Deny` eksplisit, pernyataan kebijakan ini menggunakan `Allow` dengan operator syarat Null untuk mengizinkan akses hanya ketika permintaan tidak menyertakan parameter `BypassPolicyLockoutSafetyCheck`. Ketika parameter tidak digunakan, nilai defaultnya adalah `false`. Pernyataan kebijakan yang sedikit lebih lemah ini dapat ditimpa dalam kasus yang jarang terjadi di mana bypass diperlukan.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Lihat juga

- km: KeySpec
- km: KeyOrigin
- km: KeyUsage

km: CallerAccount

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:CallerAccount`	String	Bernilai tunggal	Operasi sumber daya utama KMS	Kebijakan kunci dan kebijakan IAM

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
			Operasi penyimpanan kunci kustom	

Anda dapat menggunakan kunci kondisi ini untuk mengizinkan atau menolak akses ke semua identitas (pengguna dan peran) dalam file. Akun AWS Dalam kebijakan kunci, Anda menggunakan elemen `Principal` untuk menentukan identitas yang berlaku untuk pernyataan kebijakan. Sintaksis untuk elemen `Principal` tidak memberikan cara untuk menentukan semua identitas dalam Akun AWS. Tetapi Anda dapat mencapai efek ini dengan menggabungkan kunci kondisi ini dengan `Principal` elemen yang menentukan semua AWS identitas.

Anda dapat menggunakannya untuk mengontrol akses ke operasi sumber daya kunci KMS apa pun, yaitu AWS KMS operasi apa pun yang menggunakan kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi. Ini juga berlaku untuk operasi yang mengelola penyimpanan kunci kustom.

Misalnya, pernyataan kebijakan kunci berikut menunjukkan cara menggunakan kunci syarat `kms:CallerAccount`. Pernyataan kebijakan ini ada dalam kebijakan utama Kunci yang dikelola AWS untuk Amazon EBS. Ini menggabungkan `Principal` elemen yang menentukan semua AWS identitas dengan kunci `kms:CallerAccount` kondisi untuk secara efektif memungkinkan akses ke semua identitas di 111122223333. Akun AWS Ini berisi kunci AWS KMS kondisi tambahan (`kms:ViaService`) untuk membatasi izin lebih lanjut dengan hanya mengizinkan permintaan yang datang melalui Amazon EBS. Untuk informasi selengkapnya, lihat 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 (usang)

Kunci `kms:CustomerMasterKeySpec` kondisi tidak digunakan lagi. Sebagai gantinya, gunakan kms: KeySpec condition key.

Kunci `kms:CustomerMasterKeySpec` dan `kms:KeySpec` kondisi bekerja dengan cara yang sama. Hanya nama-nama yang berbeda. Kami menyarankan Anda menggunakan`kms:KeySpec`. Namun, untuk menghindari perubahan yang melanggar, AWS KMS mendukung kedua tombol kondisi.

kms: CustomerMasterKeyUsage (usang)

Kunci `kms:CustomerMasterKeyUsage` kondisi tidak digunakan lagi. Sebagai gantinya, gunakan kms: KeyUsage condition key.

Kunci `kms:CustomerMasterKeyUsage` dan `kms:KeyUsage` kondisi bekerja dengan cara yang sama. Hanya nama-nama yang berbeda. Kami menyarankan Anda menggunakan`kms:KeyUsage`. Namun, untuk menghindari perubahan yang melanggar, AWS KMS mendukung kedua tombol kondisi.

km: DataKeyPairSpec

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:DataKeyPairSpec`	String	Bernilai tunggal	`GenerateDataKeyPair`
			
			`GenerateDataKeyPai`
				Kebijakan kunci dan kebijakan IAM

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
			`rWithoutP laintext`	

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintextoperasi berdasarkan nilai `KeyPairSpec` parameter dalam permintaan. Misalnya, Anda dapat mengizinkan pengguna untuk menghasilkan hanya tipe tertentu dari pasangan kunci data.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:DataKeyPairSpec` kondisi untuk memungkinkan pengguna menggunakan kunci KMS untuk menghasilkan hanya pasangan kunci data RSA.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Lihat juga

- km: KeySpec
- the section called “km: EncryptionAlgorithm”
- the section called “kms:EncryptionContext: kunci-konteks”
- the section called “km: EncryptionContextKeys”

km: EncryptionAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:EncryptionAlgorithm`	String	Bernilai tunggal	`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci syarat `kms:EncryptionAlgorithm` untuk mengontrol akses ke operasi kriptografi berdasarkan algoritme enkripsi yang digunakan dalam operasi. Untuk Enkripsi, Dekripsi, dan ReEncryptoperasi, ia mengontrol akses berdasarkan nilai EncryptionAlgorithmparameter dalam permintaan. Untuk operasi yang menghasilkan kunci data dan pasangan kunci data, ini mengontrol akses berdasarkan algoritme enkripsi yang digunakan untuk mengenkripsi kunci data.

Kunci kondisi ini tidak berpengaruh pada operasi yang dilakukan di luar AWS KMS, seperti mengenkripsi dengan kunci publik dalam key pair KMS asimetris di luar. AWS KMS

EncryptionAlgorithm parameter dalam permintaan

Untuk memungkinkan pengguna hanya menggunakan algoritma enkripsi tertentu dengan kunci KMS, gunakan pernyataan kebijakan dengan `Deny` efek dan operator `StringNotEquals` kondisi. Misalnya, contoh pernyataan kebijakan kunci berikut melarang prinsipal yang dapat mengambil `ExampleRole` peran dari menggunakan kunci KMS ini dalam operasi kriptografi tertentu kecuali algoritma enkripsi dalam permintaan adalah`RSAES_OAEP_SHA_256`, algoritma enkripsi asimetris yang digunakan dengan kunci RSA KMS.

Tidak seperti pernyataan kebijakan yang memungkinkan pengguna untuk menggunakan algoritma enkripsi tertentu, pernyataan kebijakan dengan negatif ganda seperti ini mencegah kebijakan dan hibah lain untuk kunci KMS ini memungkinkan peran ini menggunakan algoritme enkripsi lainnya. `Deny`Dalam pernyataan kebijakan utama ini lebih diutamakan daripada kebijakan utama atau kebijakan IAM apa pun yang `Allow` berpengaruh, dan diutamakan daripada semua hibah untuk kunci KMS ini dan prinsipalnya.

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

Algoritma enkripsi yang digunakan untuk operasi

Anda juga dapat menggunakan kunci syarat `kms:EncryptionAlgorithm` untuk mengontrol akses ke operasi berdasarkan algoritme enkripsi yang digunakan dalam operasi, bahkan ketika algoritme tidak ditentukan dalam permintaan. Ini memungkinkan Anda untuk meminta atau melarang algoritme `SYMMETRIC_DEFAULT`, yang mungkin tidak ditentukan dalam permintaan karena itu adalah nilai default.

Fitur ini memungkinkan Anda menggunakan kunci syarat `kms:EncryptionAlgorithm` untuk mengontrol akses ke operasi yang menghasilkan kunci data dan pasangan kunci data. Operasi ini hanya menggunakan kunci KMS enkripsi simetris dan algoritma. `SYMMETRIC_DEFAULT`

Misalnya, kebijakan IAM ini membatasi perwakilannya pada enkripsi simetris. Ini menolak akses ke kunci KMS apa pun di akun contoh untuk operasi kriptografi kecuali algoritma enkripsi yang ditentukan dalam permintaan atau digunakan dalam operasi adalah SYMMETRIC_DEFAULT. Termasuk `GenerateDataKey*` menambahkan GenerateDataKey, GenerateDataKeyWithoutPlaintextGenerateDataKeyPair,, dan GenerateDataKeyPairWithoutPlaintextke izin. Syarat tidak berpengaruh pada operasi ini karena selalu menggunakan algoritme enkripsi simetris.

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

Lihat juga

- the section called “km: MacAlgorithm”
- km: SigningAlgorithm

kms:EncryptionContext: kunci-konteks

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:EncryptionContext: context-key`	String	Bernilai tunggal	`CreateGrant`
			`Encrypt`
			`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol `kms:EncryptionContext:context-key` kondisi untuk mengontrol akses ke kunci KMS enkripsi simetris berdasarkan konteks enkripsi dalam permintaan untuk operasi kriptografi. Gunakan kunci syarat ini untuk mengevaluasi kunci dan nilai dalam pasangan konteks enkripsi. Untuk mengevaluasi hanya kunci konteks enkripsi atau memerlukan konteks enkripsi terlepas dari kunci atau nilai, gunakan kms: EncryptionContextKeys condition key.

> **Note**
>
> Nilai kunci kondisi harus sesuai dengan aturan karakter untuk kebijakan utama dan kebijakan IAM. Beberapa karakter yang valid dalam konteks enkripsi tidak valid dalam kebijakan. Anda mungkin tidak dapat menggunakan kunci kondisi ini untuk mengekspresikan semua nilai konteks enkripsi yang valid. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama. Untuk detail tentang aturan dokumen kebijakan IAM, lihat Persyaratan nama IAM di Panduan Pengguna IAM.

Anda tidak dapat menentukan konteks enkripsi dalam operasi kriptografi dengan kunci KMS asimetris atau kunci KMS HMAC. Algoritma asimetris dan algoritma MAC tidak mendukung konteks enkripsi.

Untuk menggunakan kunci kondisi kunci-konteks kmsEncryptionContext::, ganti placeholder kunci-konteks dengan *`kunci konteks enkripsi`*. Ganti placeholder *`context-value`* dengan nilai konteks enkripsi.

```
"kms:EncryptionContext:context-key": "context-value"
```

Misalnya, kunci syarat berikut menentukan konteks enkripsi di mana kuncinya adalah `AppName` dan nilainya adalah `ExampleApp` (`AppName = ExampleApp`).

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Ini adalah kunci syarat bernilai tunggal. Kunci dalam kunci syarat menentukan kunci konteks enkripsi tertentu (context-key). Meskipun Anda dapat menyertakan beberapa pasangan konteks enkripsi di setiap permintaan API, pasangan konteks enkripsi dengan context-key yang ditentukan hanya dapat memiliki satu nilai. Misalnya, kunci `kms:EncryptionContext:Department` kondisi hanya berlaku untuk pasangan konteks enkripsi dengan `Department` kunci, dan setiap pasangan konteks enkripsi yang diberikan dengan `Department` kunci hanya dapat memiliki satu nilai.

Jangan gunakan operator set dengan kunci syarat `kms:EncryptionContext:`*`context-key`*. Jika Anda membuat pernyataan kebijakan dengan tindakan `Allow`, kunci syarat `kms:EncryptionContext:`*`context-key`*, dan operator set `ForAllValues`, syaratmemungkinkan permintaan tanpa konteks enkripsi dan permintaan dengan pasangan konteks enkripsi yang tidak ditentukan dalam syarat kebijakan.

> **Warning**
>
> Jangan gunakan operator set `ForAnyValue` atau `ForAllValues` dengan kunci syarat bernilai tunggal ini. Operator set ini dapat membuat syarat kebijakan yang tidak memerlukan nilai yang ingin Anda wajibkan dan mengizinkan nilai yang ingin Anda larang.
> Jika Anda membuat atau memperbarui kebijakan yang menyertakan operator `ForAllValues` set dengan kms:EncryptionContext: context-key, akan AWS KMS menampilkan pesan galat berikut: `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.`

Untuk mewajibkan pasangan konteks enkripsi tertentu, gunakan kunci syarat `kms:EncryptionContext:`*`context-key`* dengan operator `StringEquals`.

Contoh pernyataan kebijakan kunci berikut memungkinkan prinsipal yang dapat mengambil peran untuk menggunakan kunci KMS dalam `GenerateDataKey` permintaan hanya ketika konteks enkripsi dalam permintaan menyertakan pasangan. `AppName:ExampleApp` Pasangan konteks enkripsi lainnya diizinkan.

Nama kunci tidak peka huruf besar/kecil. Kepekaan terharap huruf besar/kecil dari nilai ditentukan oleh operator syarat, seperti `StringEquals`. Untuk detail selengkapnya, lihat Kepekaan terhadap huruf besar/kecil dari syarat konteks enkripsi.

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

Untuk meminta pasangan konteks enkripsi dan melarang semua pasangan konteks enkripsi lainnya, gunakan kms:EncryptionContext: context-key dan `kms:EncryptionContextKeys`dalam pernyataan kebijakan. Pernyataan kebijakan kunci berikut menggunakan syarat `kms:EncryptionContext:AppName` untuk mewajibkan pasangan konteks enkripsi `AppName=ExampleApp` dalam permintaan. Pernyataan tersebut juga menggunakan kunci syarat `kms:EncryptionContextKeys` dengan operator set `ForAllValues` untuk mengizinkan hanya kunci konteks enkripsi `AppName`.

Operator set `ForAllValues` membatasi kunci konteks enkripsi dalam permintaan ke `AppName`. Jika syarat `kms:EncryptionContextKeys` dengan operator set `ForAllValues` digunakan sendiri dalam pernyataan kebijakan, operator set ini akan mengizinkan permintaan tanpa konteks enkripsi. Namun, jika permintaan tidak memiliki konteks enkripsi, kondisi `kms:EncryptionContext:AppName` akan gagal. Untuk detail tentang operator set `ForAllValues`, lihat Menggunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KeyUsers"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "AppName"
      ]
    }
  }
}
```

Anda juga dapat menggunakan tombol kondisi ini untuk menolak akses ke kunci KMS untuk operasi tertentu. Contoh pernyataan kebijakan kunci berikut menggunakan `Deny` efek untuk melarang prinsipal menggunakan kunci KMS jika konteks enkripsi dalam permintaan menyertakan pasangan konteks `Stage=Restricted` enkripsi. Syarat ini memungkinkan permintaan dengan pasangan konteks enkripsi lainnya, termasuk pasangan konteks enkripsi dengan kunci `Stage` dan nilai lainnya, seperti `Stage=Test`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

Menggunakan beberapa pasangan konteks enkripsi

Anda dapat mewajibkan atau melarang beberapa pasangan konteks enkripsi. Anda juga dapat mewajibkan salah satu dari beberapa pasangan konteks enkripsi. Untuk detail tentang logika yang digunakan untuk menafsirkan kondisi ini, lihat Membuat syarat dengan beberapa kunci atau nilai di Panduan Pengguna IAM.

> **Note**
>
> Versi sebelumnya dari topik ini menampilkan pernyataan kebijakan yang menggunakan `ForAnyValue` dan `ForAllValues` menetapkan operator dengan kms:EncryptionContext: context-key condition key. Menggunakan operator set dengan kunci syarat bernilai tunggal dapat menghasilkan kebijakan yang mengizinkan permintaan tanpa konteks enkripsi dan pasangan konteks enkripsi yang tidak ditentukan.
> Misalnya, kondisi kebijakan dengan efek `Allow`, operator set `ForAllValues`, dan kunci syarat `"kms:EncryptionContext:Department": "IT"` tidak membatasi konteks enkripsi ke pasangan "Department=IT". Ini memungkinkan permintaan tanpa konteks enkripsi dan permintaan dengan pasangan konteks enkripsi yang tidak ditentukan, seperti `Stage=Restricted`.
> Harap tinjau kebijakan Anda dan hilangkan operator yang ditetapkan dari kondisi apa pun dengan kms:EncryptionContext: context-key. Upaya untuk membuat atau memperbarui kebijakan dengan format ini gagal dengan pengecualian `OverlyPermissiveCondition`. Untuk mengatasi kesalahan, hapus operator set.

Untuk mewajibkan beberapa pasangan konteks enkripsi, daftarkan pasangan dalam syarat yang sama. Contoh pernyataan kebijakan kunci berikut memerlukan dua pasangan konteks enkripsi, `Department=IT` dan `Project=Alpha`. Karena syarat memiliki kunci (`kms:EncryptionContext:Department` dan `kms:EncryptionContext:Project`) yang berbeda, syarat tersebut secara implisit dihubungkan oleh operator DAN. Pasangan konteks enkripsi lainnya diizinkan, tetapi tidak diwajibkan.

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

Untuk meminta satu pasangan konteks enkripsi ATAU pasangan lain, tempatkan setiap kunci syarat dalam pernyataan kebijakan terpisah. Contoh kebijakan kunci berikut memerlukan pasangan `Department=IT` atau `Project=Alpha`, atau keduanya. Pasangan konteks enkripsi lainnya diizinkan, tetapi tidak diwajibkan.

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

Untuk meminta pasangan enkripsi tertentu dan mengecualikan semua pasangan konteks enkripsi lainnya, gunakan kms:EncryptionContext: context-key dan [kms:EncryptionContextKeys](#)dalam pernyataan kebijakan. Pernyataan kebijakan kunci berikut menggunakan kondisi kms:EncryptionContext: context-key untuk memerlukan konteks enkripsi dengan keduanya dan pasangan. *`Department=IT`* `Project=Alpha` Ini menggunakan kunci syarat `kms:EncryptionContextKeys` dengan operator set `ForAllValues` untuk mengizinkan hanya kunci konteks enkripsi `Department` dan `Project`.

Operator set `ForAllValues` membatasi kunci konteks enkripsi dalam permintaan ke `Department` dan `Project`. Jika digunakan sendiri dalam suatu kondisi, operator set ini akan mengizinkan permintaan tanpa konteks enkripsi, tetapi dalam konfigurasi ini, kunci konteks kms:EncryptionContext: dalam kondisi ini akan gagal.

```
{
  "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"
```

```
      ]
    }
  }
}
```

Anda juga dapat melarang beberapa pasangan konteks enkripsi. Contoh pernyataan kebijakan kunci berikut menggunakan `Deny` efek untuk melarang prinsipal menggunakan kunci KMS jika konteks enkripsi dalam permintaan menyertakan `Stage=Restricted` `Stage=Production` atau.pair.

Beberapa nilai (`Restricted` dan `Production`) untuk kunci yang sama (`kms:EncryptionContext:Stage`) secara implisit dihubungkan oleh ATAU. Untuk detailnya, lihat Logika evaluasi untuk syarat dengan beberapa kunci atau nilai di Panduan Pengguna IAM.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
         "Restricted",
         "Production"
      ]
    }
  }
}
```

Kepekaan terhadap huruf besar/kecil dari syarat konteks enkripsi

Konteks enkripsi yang ditentukan dalam operasi dekripsi harus sama persis, peka huruf besar/kecil untuk konteks enkripsi yang ditentukan dalam operasi enkripsi. Hanya urutan pasangan dalam konteks enkripsi dengan banyak pasangan yang dapat bervariasi.

Namun, dalam kondisi kebijakan, kunci syarat tidak peka huruf besar/kecil. Pengaturan kepekaan terhadap huruf besar/kecil dari nilai syarat ditentukan oleh operator syarat kebijakan yang Anda gunakan, seperti `StringEquals` atau `StringEqualsIgnoreCase`.

Dengan demikian, kunci syarat, yang terdiri dari awalan `kms:EncryptionContext:` dan pengganti *`context-key`*, tidak peka huruf besar/kecil. Kebijakan yang menggunakan syarat ini tidak

memeriksa huruf salah satu elemen kunci syarat. Kepekaan terhadap huruf besar/kecil dari nilai, yaitu, penggantian *`context-value`*, ditentukan oleh operator syarat kebijakan.

Misalnya, pernyataan kebijakan berikut mengizinkan operasi saat konteks enkripsi menyertakan kunci `Appname`, terlepas dari kapitalisasinya. Syarat `StringEquals` mengharuskan `ExampleApp` dikapitalisasi seperti yang ditentukan.

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

Untuk mewajibkan kunci konteks enkripsi peka huruf besar/kecil, gunakan kondisi EncryptionContextKeys kebijakan kms: dengan operator kondisi peka huruf besar/kecil, seperti. `StringEquals` Dalam syarat kebijakan ini, karena kunci konteks enkripsi adalah nilai dalam syarat kebijakan ini, kepekaan terhadap huruf besar/kecilnya ditentukan oleh operator syarat.

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

Untuk memerlukan evaluasi case-sensitive dari kunci konteks enkripsi dan nilai, gunakan kondisi kebijakan kunci-konteks *kms:EncryptionContextKeys* dan kmsEncryptionContext:: bersama-sama dalam pernyataan kebijakan yang sama. Operator syarat peka huruf besar/kecil (seperti `StringEquals`) selalu berlaku untuk nilai syarat. Kunci konteks enkripsi (seperti `AppName`) adalah nilai syarat `kms:EncryptionContextKeys`. Nilai konteks enkripsi (sepertiExampleApp) adalah nilai dari kondisi kunci-konteks kmsEncryptionContext::.

Misalnya, dalam contoh pernyataan kebijakan kunci berikut, karena operator `StringEquals` peka huruf besar/kecil, baik kunci konteks enkripsi maupun nilai konteks enkripsi peka huruf besar/kecil.

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

Menggunakan variabel dalam syarat konteks enkripsi

Kunci dan nilai dalam pasangan konteks enkripsi harus berupa string literal sederhana. Kunci dan nilai tersebut tidak bisa berupa bilangan bulat atau objek, atau jenis apa pun yang tidak sepenuhnya diselesaikan. Jika Anda menggunakan tipe yang berbeda, seperti integer atau float, AWS KMS menafsirkannya sebagai string literal.

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

Namun, nilai kunci syarat `kms:EncryptionContext:`*`context-key`* dapat berupa variabel kebijakan IAM. Variabel kebijakan ini diselesaikan saat runtime berdasarkan nilai dalam permintaan.

Misalnya, `aws:CurrentTime` memutuskan waktu permintaan dan `aws:username` memutuskan nama ramah pemanggil.

Anda dapat menggunakan variabel kebijakan ini untuk membuat pernyataan kebijakan dengan syarat yang memerlukan informasi yang sangat spesifik dalam konteks enkripsi, seperti nama pengguna pemanggil. Karena berisi variabel, Anda dapat menggunakan pernyataan kebijakan yang sama untuk semua pengguna yang dapat mengambil peran tersebut. Anda tidak perlu menulis pernyataan kebijakan terpisah untuk setiap pengguna.

Pertimbangkan situasi di mana Anda ingin semua pengguna yang dapat mengambil peran untuk menggunakan kunci KMS yang sama untuk mengenkripsi dan mendekripsi data mereka. Namun, Anda ingin mengizinkan mereka mendekripsi hanya data yang mereka enkripsi. Mulailah dengan mengharuskan setiap permintaan untuk AWS KMS menyertakan konteks enkripsi di mana kuncinya berada `user` dan nilainya adalah nama AWS pengguna pemanggil, seperti yang berikut.

```
"encryptionContext": {
    "user": "bob"
}
```

Kemudian, untuk menerapkan persyaratan ini, Anda dapat menggunakan pernyataan kebijakan seperti pada contoh berikut. Pernyataan kebijakan ini memberikan izin `TestTeam` peran untuk mengenkripsi dan mendekripsi data dengan kunci KMS. Namun, izin hanya valid jika konteks enkripsi dalam permintaan menyertakan pasangan `"user": "`*`<username>`*`"`. Untuk mewakili nama pengguna, kondisi menggunakan variabel kebijakan `aws:username`.

Saat permintaan dievaluasi, nama pengguna pemanggil menggantikan variabel dalam syarat. Dengan begitu, syarat memerlukan konteks enkripsi `"user": "bob"` untuk "bob" dan `"user": "alice"` untuk "alice."

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
```

```
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

Anda dapat menggunakan variabel kebijakan IAM hanya dalam nilai kunci syarat `kms:EncryptionContext:`*`context-key`*. Anda tidak dapat menggunakan variabel dalam kunci.

Anda juga dapat menggunakan kunci konteks khusus penyedia dalam variabel. Kunci konteks ini secara unik mengidentifikasi pengguna yang masuk AWS dengan menggunakan federasi identitas web.

Seperti semua variabel, variabel ini hanya dapat digunakan dalam syarat kebijakan `kms:EncryptionContext:`*`context-key`*, bukan dalam konteks enkripsi yang sebenarnya. Dan variable hanya dapat digunakan dalam nilai syarat, bukan di kunci.

Misalnya, pernyataan kebijakan kunci berikut ini mirip dengan kebijakan kunci sebelumnya. Namun, kondisi tersebut memerlukan konteks enkripsi di mana kuncinya adalah `sub` dan nilainya secara unik mengidentifikasi pengguna yang masuk ke kumpulan pengguna Amazon Cognito. Untuk detail tentang mengidentifikasi pengguna dan peran di Amazon Cognito, lihat Peran IAM di Panduan Developer Amazon Cognito.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
       "kms:EncryptionContext:sub": "${cognito-identity.amazonaws.com:sub}"
    }
  }
}
```

Lihat juga

- the section called “km: EncryptionContextKeys”
- the section called “km: GrantConstraintType”

km: EncryptionContextKeys

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:EncryptionContextKeys`	String (daftar)	Multi-nilai	`CreateGrant`
			`Decrypt`
			`Encrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
			`ReEncrypt`
				Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol `kms:EncryptionContextKeys` kondisi untuk mengontrol akses ke kunci KMS enkripsi simetris berdasarkan konteks enkripsi dalam permintaan untuk operasi kriptografi. Gunakan kunci syarat ini untuk mengevaluasi hanya kunci di setiap pasangan konteks enkripsi. Untuk mengevaluasi kunci dan nilai dalam konteks enkripsi, gunakan kunci syarat `kms:EncryptionContext:`*`context-key`*.

Anda tidak dapat menentukan konteks enkripsi dalam operasi kriptografi dengan kunci KMS asimetris atau kunci KMS HMAC. Algoritma asimetris dan algoritma MAC tidak mendukung konteks enkripsi.

> **Note**
>
> Nilai kunci kondisi, termasuk kunci konteks enkripsi, harus sesuai dengan karakter dan aturan pengkodean untuk kebijakan AWS KMS kunci. Anda mungkin tidak dapat menggunakan kunci kondisi ini untuk mengekspresikan semua kunci konteks enkripsi yang valid. Untuk detail tentang aturan dokumen kebijakan utama, lihatFormat kebijakan utama. Untuk detail tentang aturan dokumen kebijakan IAM, lihat Persyaratan nama IAM di Panduan Pengguna IAM.

Ini adalah kunci syarat multi-nilai. Anda dapat menentukan beberapa pasangan konteks enkripsi di setiap permintaan API. `kms:EncryptionContextKeys`membandingkan kunci konteks enkripsi dalam permintaan dengan kumpulan kunci konteks enkripsi dalam kebijakan. Untuk menentukan bagaimana set ini dibandingkan, Anda harus menyediakan operator set `ForAnyValue` atau `ForAllValues` dalam syarat kebijakan. Untuk detail tentang operator kumpulan, lihat Menggunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

- `ForAnyValue`: Setidaknya satu kunci konteks enkripsi dalam permintaan harus cocok dengan kunci konteks enkripsi dalam syarat kebijakan. Kunci konteks enkripsi lainnya diizinkan. Jika permintaan tidak memiliki konteks enkripsi, syaratnya tidak terpenuhi.
- `ForAllValues`: Setiap kunci konteks enkripsi dalam permintaan harus cocok dengan kunci konteks enkripsi dalam syarat kebijakan. Operator set ini membatasi kunci konteks enkripsi untuk yang ada dalam syarat kebijakan. Operator set ini tidak mewajibkan kunci konteks enkripsi apa pun, tetapi melarang kunci konteks enkripsi yang tidak ditentukan.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci syarat `kms:EncryptionContextKeys` dengan operator set `ForAnyValue`. Pernyataan kebijakan ini memungkinkan penggunaan kunci KMS untuk operasi yang ditentukan, tetapi hanya jika setidaknya satu dari pasangan konteks enkripsi dalam permintaan menyertakan `AppName` kunci, terlepas dari nilainya.

Misalnya, pernyataan kebijakan kunci ini mengizinkan permintaan `GenerateDataKey` dengan dua pasangan konteks enkripsi, `AppName=Helper` dan `Project=Alpha`, karena pasangan konteks

enkripsi pertama memenuhi syarat. Permintaan dengan hanya `Project=Alpha` atau tanpa konteks enkripsi akan gagal.

Karena operasi StringEqualskondisi peka huruf besar/kecil, pernyataan kebijakan ini memerlukan ejaan dan kasus kunci konteks enkripsi. Tetapi Anda dapat menggunakan operator syarat yang mengabaikan huruf besar/kecil kunci, seperti `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Anda juga dapat menggunakan kunci `kms:EncryptionContextKeys` kondisi untuk memerlukan konteks enkripsi (konteks enkripsi apa pun) dalam operasi kriptografi yang menggunakan kunci KMS;

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:EncryptionContextKeys` kondisi dengan operator kondisi Null untuk mengizinkan akses ke kunci KMS hanya jika konteks enkripsi dalam permintaan API tidak null. Syarat ini tidak memeriksa kunci atau nilai konteks enkripsi. Syarat ini hanya memverifikasi bahwa konteks enkripsi ada.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
```

```
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

Lihat juga

- kms:EncryptionContext: kunci-konteks
- km: GrantConstraintType

km: ExpirationModel

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ExpirationModel`	String	Bernilai tunggal	`ImportKeyMaterial`	Kebijakan kunci dan kebijakan IAM

Kunci `kms:ExpirationModel` kondisi mengontrol akses ke ImportKeyMaterialoperasi berdasarkan nilai ExpirationModelparameter dalam permintaan.

`ExpirationModel` adalah parameter opsional yang menentukan apakah material kunci yang diimpor kedaluwarsa. Nilai yang valid adalah `KEY_MATERIAL_EXPIRES` dan `KEY_MATERIAL_DOES_NOT_EXPIRE`. `KEY_MATERIAL_EXPIRES` adalah nilai default.

Tanggal dan waktu kedaluwarsa ditentukan oleh nilai parameter. ValidTo Parameter `ValidTo` diperlukan kecuali nilai parameter `ExpirationModel` adalah `KEY_MATERIAL_DOES_NOT_EXPIRE`. Anda juga dapat menggunakan kms: ValidTo condition key untuk meminta tanggal kedaluwarsa tertentu sebagai syarat untuk akses.

Contoh pernyataan kebijakan berikut menggunakan kunci `kms:ExpirationModel` kondisi untuk memungkinkan pengguna mengimpor materi kunci ke kunci KMS hanya jika permintaan menyertakan `ExpirationModel` parameter dan nilainya. `KEY_MATERIAL_DOES_NOT_EXPIRE`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

Anda juga dapat menggunakan tombol `kms:ExpirationModel` kondisi untuk memungkinkan pengguna mengimpor materi kunci hanya ketika materi kunci kedaluwarsa. Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:ExpirationModel` kondisi dengan operator kondisi Null untuk memungkinkan pengguna mengimpor materi kunci hanya ketika permintaan tidak memiliki `ExpirationModel` parameter. Nilai default untuk ExpirationModel adalah`KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Lihat juga

- km: ValidTo
- km: WrappingAlgorithm
- km: WrappingKeySpec

km: GrantConstraintType

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GrantConstraintType`	String	Bernilai tunggal	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan jenis batasan hibah dalam permintaan.

Saat Anda membuat pemberian izin, Anda dapat secara opsional menentukan batasan pemberian izin untuk mengizinkan operasi yang izinnya hanya diizinkan saat konteks enkripsi tertentu ada. Batasan pemberian izin dapat berupa salah satu dari dua jenis: `EncryptionContextEquals` atau `EncryptionContextSubset`. Anda dapat menggunakan kunci syarat ini untuk memeriksa apakah permintaan berisi satu jenis atau jenis lain.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:GrantConstraintType` kondisi untuk memungkinkan pengguna membuat hibah hanya jika permintaan menyertakan batasan `EncryptionContextEquals` hibah. Contoh menunjukkan pernyataan kebijakan dalam kebijakan kunci.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
```

```
    }
  }
}
```

Lihat juga

- kms:EncryptionContext: kunci-konteks
- km: EncryptionContextKeys
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: GranteePrincipal
- km: RetiringPrincipal

km: GrantIsFor AWSResource

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan

`kms:GrantIsForAWSResource`	Boolean	Bernilai tunggal	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				Kebijakan kunci dan kebijakan IAM

Mengizinkan atau menolak izin untuk CreateGrant, ListGrants, atau RevokeGrantoperasi hanya ketika AWS layanan terintegrasi dengan AWS KMS panggilan operasi atas nama pengguna. Syarat kebijakan ini tidak mengizinkan pengguna untuk memanggil operasi pemberian izin ini secara langsung.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci syarat `kms:GrantIsForAWSResource`. Ini memungkinkan AWS layanan yang terintegrasi dengan AWS KMS, seperti Amazon EBS, untuk membuat hibah pada kunci KMS ini atas nama prinsipal yang ditentukan.

```
{
  "Effect": "Allow",
  "Principal": {
```

```
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Lihat juga

- km: GrantConstraintType
- km: GrantOperations
- km: GranteePrincipal
- km: RetiringPrincipal

km: GrantOperations

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GrantOperations`	String	Multi-nilai	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan operasi hibah dalam permintaan. Misalnya, Anda dapat mengizinkan pengguna membuat hibah yang mendelegasikan izin untuk mengenkripsi tetapi tidak mendekripsi. Untuk informasi selengkapnya tentang pemberian izin, lihat Menggunakan pemberian izin.

Ini adalah kunci syarat multi-nilai. `kms:GrantOperations` membandingkan kumpulan operasi pemberian izin dalam permintaan `CreateGrant` dengan kumpulan operasi pemberian izin dalam kebijakan. Untuk menentukan bagaimana set ini dibandingkan, Anda harus menyediakan operator set `ForAnyValue` atau `ForAllValues` dalam syarat kebijakan. Untuk detail tentang operator set, lihat Menggunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

- `ForAnyValue`: Setidaknya satu operasi pemberian izin dalam permintaan harus cocok dengan salah satu operasi pemberian izin dalam syarat kebijakan. Operasi pemberian izin lainnya diizinkan.
- ForAllValues: Setiap operasi hibah dalam permintaan harus sesuai dengan operasi hibah dalam kondisi kebijakan. Operator set ini membatasi operasi pemberian izin untuk yang ditentukan dalam syarat kebijakan. Operator set ini tidak mewajibkan operasi pemberian izin apa pun, tetapi melarang operasi pemberian izin yang tidak ditentukan.

  ForAllValues juga mengembalikan true ketika tidak ada operasi hibah dalam permintaan, tetapi `CreateGrant` tidak mengizinkannya. Jika parameter `Operations` tidak ada atau memiliki nilai null, permintaan `CreateGrant` gagal.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:GrantOperations` kondisi untuk membuat hibah hanya ketika operasi hibah`Encrypt`,`ReEncryptTo`, atau keduanya. Jika pemberian izin mencakup operasi lain, permintaan `CreateGrant` gagal.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

Jika Anda mengubah operator yang ditetapkan dalam syarat kebijakan ke `ForAnyValue`, pernyataan kebijakan akan mewajibkan setidaknya salah satu operasi pemberian izin dalam izin adalah `Encrypt` atau `ReEncryptTo`, tetapi akan mengizinkan operasi pemberian izin lainnya, seperti `Decrypt` atau `ReEncryptFrom`.

Lihat juga

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GranteePrincipal
- km: RetiringPrincipal

km: GranteePrincipal

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:GranteePrincipal`	String	Bernilai tunggal	`CreateGrant`	Kebijakan IAM dan kunci

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan nilai GranteePrincipalparameter dalam permintaan. Misalnya, Anda dapat membuat hibah untuk menggunakan kunci KMS hanya jika prinsipal penerima hibah dalam `CreateGrant` permintaan cocok dengan prinsipal yang ditentukan dalam pernyataan kondisi.

Untuk menentukan pokok penerima hibah, gunakan Amazon Resource Name (ARN) dari prinsipal. AWS Prinsipal yang valid meliputi Akun AWS, pengguna IAM, peran IAM, pengguna federasi, dan pengguna peran yang diasumsikan. Untuk bantuan dengan sintaks ARN untuk prinsipal, lihat ARN IAM di Panduan Pengguna IAM.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:GranteePrincipal` kondisi untuk membuat hibah untuk kunci KMS hanya jika pokok penerima hibah dalam hibah adalah. `LimitedAdminRole`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
```

```
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Lihat juga

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: RetiringPrincipal

km: KeyOrigin

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:KeyOrigin`	String	Bernilai tunggal	`CreateKey` Operasi sumber daya utama KMS	Kebijakan IAM Kebijakan kunci dan kebijakan IAM

Kunci `kms:KeyOrigin` kondisi mengontrol akses ke operasi berdasarkan nilai `Origin` properti kunci KMS yang dibuat oleh atau digunakan dalam operasi. Kunci syarat ini berfungsi sebagai syarat sumber daya atau syarat permintaan.

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateKeyoperasi berdasarkan nilai parameter Origin dalam permintaan. Nilai yang valid untuk `Origin` adalah `AWS_KMS`, `AWS_CLOUDHSM`, dan `EXTERNAL`.

Misalnya, Anda dapat membuat kunci KMS hanya ketika materi kunci dihasilkan di AWS KMS (`AWS_KMS`), hanya ketika materi kunci dihasilkan dalam AWS CloudHSM klaster yang terkait dengan penyimpanan kunci kustom (`AWS_CLOUDHSM`), atau hanya ketika materi kunci diimpor dari sumber eksternal (`EXTERNAL`).

Contoh pernyataan kebijakan kunci berikut menggunakan kunci `kms:KeyOrigin` kondisi untuk membuat kunci KMS hanya ketika AWS KMS membuat materi kunci.

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

Anda juga dapat menggunakan tombol `kms:KeyOrigin` kondisi untuk mengontrol akses ke operasi yang menggunakan atau mengelola kunci KMS berdasarkan `Origin` properti kunci KMS yang digunakan untuk operasi. Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang

diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal untuk melakukan operasi sumber daya kunci KMS yang ditentukan, tetapi hanya dengan kunci KMS di akun yang dibuat di penyimpanan kunci kustom.

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

Lihat juga

- km: BypassPolicyLockoutSafetyCheck
- km: KeySpec
- km: KeyUsage

km: KeySpec

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:KeySpec`	String	Bernilai tunggal	`CreateKey`	Kebijakan IAM

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
			Operasi sumber daya utama KMS	Kebijakan kunci dan kebijakan IAM

Kunci `kms:KeySpec` kondisi mengontrol akses ke operasi berdasarkan nilai `KeySpec` properti kunci KMS yang dibuat oleh atau digunakan dalam operasi.

Anda dapat menggunakan kunci kondisi ini dalam kebijakan IAM untuk mengontrol akses ke CreateKeyoperasi berdasarkan nilai KeySpecparameter dalam `CreateKey` permintaan. Misalnya, Anda dapat menggunakan kondisi ini untuk memungkinkan pengguna membuat hanya kunci KMS enkripsi simetris atau hanya kunci KMS HMAC.

Contoh berikut pernyataan kebijakan IAM menggunakan kunci `kms:KeySpec` kondisi untuk memungkinkan prinsipal untuk membuat hanya kunci KMS asimetris RSA. Izin hanya berlaku ketika permintaan dimulai denganRSA_. `KeySpec`

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

Anda juga dapat menggunakan tombol `kms:KeySpec` kondisi untuk mengontrol akses ke operasi yang menggunakan atau mengelola kunci KMS berdasarkan `KeySpec` properti kunci KMS yang digunakan untuk operasi. Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal untuk melakukan operasi sumber daya kunci KMS yang ditentukan, tetapi hanya dengan kunci KMS enkripsi simetris di akun.

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

Lihat juga

- km: BypassPolicyLockoutSafetyCheck
- kms: CustomerMasterKeySpec (usang)
- km: DataKeyPairSpec
- km: KeyOrigin
- km: KeyUsage

km: KeyUsage

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:KeyUsage`	String	Bernilai tunggal	`CreateKey` Operasi sumber daya utama KMS	Kebijakan IAM Kebijakan kunci dan kebijakan IAM

Kunci `kms:KeyUsage` kondisi mengontrol akses ke operasi berdasarkan nilai `KeyUsage` properti kunci KMS yang dibuat oleh atau digunakan dalam operasi.

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateKeyoperasi berdasarkan nilai KeyUsageparameter dalam permintaan. Nilai yang valid untuk `KeyUsage` adalah `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, dan `GENERATE_VERIFY_MAC`.

Misalnya, Anda dapat membuat kunci KMS hanya jika `KeyUsage` ada `ENCRYPT_DECRYPT` atau menolak izin pengguna saat `KeyUsage` ada`SIGN_VERIFY`.

Contoh berikut pernyataan kebijakan IAM menggunakan kunci `kms:KeyUsage` kondisi untuk membuat kunci KMS hanya ketika adalah`KeyUsage`. `ENCRYPT_DECRYPT`

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

Anda juga dapat menggunakan tombol `kms:KeyUsage` kondisi untuk mengontrol akses ke operasi yang menggunakan atau mengelola kunci KMS berdasarkan `KeyUsage` properti kunci KMS dalam operasi. Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal untuk melakukan operasi sumber daya kunci KMS yang ditentukan, tetapi hanya dengan kunci KMS di akun yang digunakan untuk penandatanganan dan verifikasi.

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

Lihat juga

- km: BypassPolicyLockoutSafetyCheck
- kms: CustomerMasterKeyUsage (usang)
- km: KeyOrigin
- km: KeySpec

km: MacAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:MacAlgorithm`	String	Bernilai tunggal	`GenerateMac` `VerifyMac`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol `kms:MacAlgorithm` kondisi untuk mengontrol akses ke GenerateMacdan VerifyMacoperasi berdasarkan nilai `MacAlgorithm` parameter dalam permintaan.

Contoh kebijakan kunci berikut memungkinkan pengguna yang dapat mengambil `testers` peran untuk menggunakan kunci HMAC KMS untuk menghasilkan dan memverifikasi tag HMAC hanya ketika algoritma MAC dalam permintaan adalah atau. `HMAC_SHA_384` `HMAC_SHA_512` Kebijakan ini menggunakan dua pernyataan kebijakan terpisah masing-masing dengan kondisinya sendiri. Jika Anda menentukan lebih dari satu algoritma MAC dalam pernyataan kondisi tunggal, kondisi memerlukan kedua algoritma, bukan satu atau yang lain.

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

Lihat juga

- the section called “km: EncryptionAlgorithm”
- km: SigningAlgorithm

km: MessageType

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:MessageType`	String	Bernilai tunggal	`Sign` `Verify`	Kebijakan kunci dan kebijakan IAM

Kunci syarat `kms:MessageType` mengontrol akses ke operasi Tandatangani dan Verifikasi berdasarkan nilai parameter `MessageType` dalam permintaan. Nilai yang valid untuk `MessageType` adalah `RAW` dan `DIGEST`.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:MessageType` kondisi untuk menggunakan kunci KMS asimetris untuk menandatangani pesan, tetapi bukan intisari pesan.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:MessageType": "RAW"
    }
  }
}
```

Lihat juga

- the section called “km: SigningAlgorithm”

km: MultiRegion

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:MultiRegion`	Boolean	Bernilai tunggal	`CreateKey` Operasi sumber daya utama KMS	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci syarat ini untuk mengizinkan operasi hanya pada kunci Wilayah tunggal atau hanya pada kunci multi-Wilayah. Kunci `kms:MultiRegion` kondisi mengontrol akses ke AWS KMS operasi pada kunci KMS dan CreateKeyoperasi berdasarkan nilai `MultiRegion` properti kunci KMS. Nilai yang valid adalah `true` (multi-Wilayah), dan `false` (Wilayah tunggal). Semua kunci KMS memiliki `MultiRegion` properti.

Misalnya, pernyataan kebijakan IAM berikut menggunakan kunci syarat `kms:MultiRegion` untuk mengizinkan perwakilan membuat kunci Wilayah tunggal saja.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}
```

km: MultiRegionKeyType

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:MultiRegionKeyType`	String	Bernilai tunggal	`CreateKey`	Kebijakan kunci dan kebijakan IAM

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
			Operasi sumber daya utama KMS	

Anda dapat menggunakan kunci syarat ini untuk mengizinkan operasi hanya pada kunci utama multi-Wilayah atau hanya pada kunci replika multi-Wilayah. Kunci `kms:MultiRegionKeyType` kondisi mengontrol akses ke AWS KMS operasi pada kunci KMS dan CreateKeyoperasi berdasarkan `MultiRegionKeyType` properti kunci KMS. Nilai yang valid adalah `PRIMARY` dan `REPLICA`. Hanya kunci Multi-region yang memiliki `MultiRegionKeyType` properti.

Biasanya, Anda menggunakan kunci `kms:MultiRegionKeyType` kondisi dalam kebijakan IAM untuk mengontrol akses ke beberapa kunci KMS. Namun, karena kunci Multi-wilayah tertentu dapat berubah menjadi primer atau replika, Anda mungkin ingin menggunakan kondisi ini dalam kebijakan kunci untuk mengizinkan operasi hanya jika kunci Multi-wilayah tertentu adalah kunci primer atau replika.

Misalnya, pernyataan kebijakan IAM berikut menggunakan kunci syarat `kms:MultiRegionKeyType` untuk mengizinkan perwakilan menjadwalkan dan membatalkan penghapusan kunci hanya pada kunci replika multi-Wilayah di Akun AWS yang ditentukan.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}
```

Untuk mengizinkan atau menolak akses ke semua kunci multi-Wilayah, Anda bisa menggunakan kedua nilai atau nilai null dengan `kms:MultiRegionKeyType`. Namun, kunci MultiRegion kondisi kms: direkomendasikan untuk tujuan itu.

km: PrimaryRegion

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:PrimaryRegion`	String (daftar)	Bernilai tunggal	`UpdatePrimaryRegion`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk membatasi Wilayah tujuan dalam suatu UpdatePrimaryRegionoperasi. Ini adalah Wilayah AWS yang dapat meng-host kunci utama Multi-wilayah Anda.

Kunci `kms:PrimaryRegion` kondisi mengontrol akses ke UpdatePrimaryRegionoperasi berdasarkan nilai `PrimaryRegion` parameter. `PrimaryRegion`Parameter menentukan kunci replika Multi-region yang sedang dipromosikan ke primer. Wilayah AWS Nilai kondisi adalah satu atau lebih Wilayah AWS nama, seperti `us-east-1` atau`ap-southeast-2`, atau pola nama Wilayah, seperti `eu-*`

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:PrimaryRegion` kondisi untuk mengizinkan prinsipal memperbarui wilayah utama kunci Multi-wilayah ke salah satu dari empat Wilayah yang ditentukan.

```
{
  "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 kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ReEncryptOnSameKey`	Boolean	Bernilai tunggal	`ReEncrypt`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk mengontrol akses ke ReEncryptoperasi berdasarkan apakah permintaan menentukan kunci KMS tujuan yang sama dengan yang digunakan untuk enkripsi asli.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:ReEncryptOnSameKey` kondisi untuk mengenkripsi ulang hanya jika kunci KMS tujuan sama dengan yang digunakan untuk enkripsi asli.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}
```

km: RequestAlias

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:RequestAlias`	String (daftar)	Bernilai tunggal	Operasi kriptografi DescribeKey GetPublicKey	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk mengizinkan operasi hanya ketika permintaan menggunakan alias tertentu untuk mengidentifikasi kunci KMS. Kunci `kms:RequestAlias` kondisi mengontrol akses ke kunci KMS yang digunakan dalam operasi kriptografi,`GetPublicKey`, atau `DescribeKey` berdasarkan alias yang mengidentifikasi kunci KMS dalam permintaan. (Kondisi kebijakan ini tidak berpengaruh pada GenerateRandomoperasi karena operasi tidak menggunakan kunci KMS atau alias.)

Kondisi ini mendukung kontrol akses berbasis atribut (ABAC) di AWS KMS, yang memungkinkan Anda mengontrol akses ke kunci KMS berdasarkan tag dan alias kunci KMS. Anda dapat menggunakan tag dan alias untuk mengizinkan atau menolak akses ke kunci KMS tanpa mengubah kebijakan atau hibah. Untuk detail selengkapnya, lihat ABAC untuk AWS KMS.

Untuk menentukan alias dalam syarat kebijakan ini, gunakan nama alias, seperti `alias/project-alpha`, atau pola nama alias, seperti `alias/*test*`. Anda tidak dapat menentukan alias ARN dalam nilai kunci syarat ini.

Untuk memenuhi syarat ini, nilai parameter `KeyId` dalam permintaan harus berupa nama alias yang cocok atau alias ARN. Jika permintaan menggunakan pengidentifikasi kunci yang berbeda, itu tidak memenuhi kondisi, bahkan jika mengidentifikasi kunci KMS yang sama.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan prinsipal untuk memanggil GenerateDataKeyoperasi pada kunci KMS. Namun ini hanya diizinkan jika nilai parameter `KeyId` dalam permintaan adalah `alias/finance-key` atau alias ARN dengan nama alias itu, seperti `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"
    }
  }
}
```

Anda tidak dapat menggunakan tombol kondisi ini untuk mengontrol akses ke operasi alias, seperti CreateAliasatau DeleteAlias. Untuk informasi tentang mengontrol akses ke operasi alias, lihat Mengontrol akses ke alias.

km: ResourceAliases

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ResourceAliases`	String (daftar)	Multi-nilai	Operasi sumber daya utama KMS	Kebijakan IAM saja

Gunakan tombol kondisi ini untuk mengontrol akses ke kunci KMS berdasarkan alias yang terkait dengan kunci KMS. Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi.

Kondisi ini mendukung kontrol akses berbasis atribut (ABAC) di. AWS KMS Dengan ABAC, Anda dapat mengontrol akses ke kunci KMS berdasarkan tag yang ditetapkan ke kunci KMS dan alias yang terkait dengan kunci KMS. Anda dapat menggunakan tag dan alias untuk mengizinkan atau menolak akses ke kunci KMS tanpa mengubah kebijakan atau hibah. Lihat perinciannya di ABAC untuk AWS KMS.

Alias harus unik di Akun AWS dan Wilayah, tetapi kondisi ini memungkinkan Anda mengontrol akses ke beberapa kunci KMS di Wilayah yang sama (menggunakan operator `StringLike` perbandingan) atau ke beberapa kunci KMS di setiap akun yang berbeda Wilayah AWS .

> **Note**
>
> ResourceAliasesKondisi kms: hanya efektif jika kunci KMS sesuai dengan alias per kuota kunci KMS. Jika kunci KMS melebihi kuota ini, kepala sekolah yang berwenang untuk menggunakan kunci KMS dengan `kms:ResourceAliases` syarat ditolak akses ke kunci KMS.

Untuk menentukan alias dalam syarat kebijakan ini, gunakan nama alias, seperti `alias/project-alpha`, atau pola nama alias, seperti `alias/*test*`. Anda tidak dapat menentukan alias ARN dalam nilai kunci syarat ini. Untuk memenuhi kondisi, kunci KMS yang digunakan dalam operasi harus memiliki alias yang ditentukan. Tidak masalah apakah atau bagaimana kunci KMS diidentifikasi dalam permintaan operasi.

Ini adalah kunci kondisi multivaluasi yang membandingkan kumpulan alias yang terkait dengan kunci KMS dengan kumpulan alias dalam kebijakan. Untuk menentukan bagaimana set ini dibandingkan, Anda harus menyediakan operator set `ForAnyValue` atau `ForAllValues` dalam syarat kebijakan. Untuk detail tentang operator kumpulan, lihat Menggunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

- ForAnyValue: Setidaknya satu alias yang terkait dengan kunci KMS harus cocok dengan alias dalam kondisi kebijakan. Alias lainnya diizinkan. Jika kunci KMS tidak memiliki alias, kondisinya tidak terpenuhi.
- ForAllValues: Setiap alias yang terkait dengan kunci KMS harus cocok dengan alias dalam kebijakan. Operator set ini membatasi alias yang terkait dengan kunci KMS ke alias yang berada dalam kondisi kebijakan. Operator set ini tidak mewajibkan alias apa pun, tetapi melarang alias yang tidak ditentukan.

Misalnya, pernyataan kebijakan IAM berikut memungkinkan prinsipal untuk memanggil GenerateDataKeyoperasi pada setiap kunci KMS dalam Akun AWS yang ditentukan yang terkait dengan alias. `finance-key` (Kebijakan kunci kunci KMS yang terpengaruh juga harus mengizinkan akun prinsipal untuk menggunakannya untuk operasi ini.) Untuk menunjukkan bahwa kondisi terpenuhi ketika salah satu dari banyak alias yang mungkin terkait dengan kunci KMS adalah`alias/finance-key`, kondisi menggunakan operator `ForAnyValue` set.

Karena `kms:ResourceAliases` kondisi didasarkan pada sumber daya, bukan permintaan, panggilan untuk `GenerateDataKey` berhasil untuk setiap kunci KMS yang terkait dengan `finance-`

key alias, bahkan jika permintaan menggunakan ID kunci atau kunci ARN untuk mengidentifikasi kunci KMS.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": "kms:GenerateDataKey",
  "Resource": [
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:ResourceAliases": "alias/finance-key"
    }
  }
}
```

Contoh berikut pernyataan kebijakan IAM memungkinkan prinsipal untuk mengaktifkan dan menonaktifkan kunci KMS tetapi hanya ketika semua alias kunci KMS menyertakan "." `Test` Pernyataan kebijakan ini menggunakan dua kondisi. Kondisi dengan operator `ForAllValues` set mengharuskan semua alias yang terkait dengan kunci KMS menyertakan “Uji”. Kondisi dengan operator yang `ForAnyValue` ditetapkan mengharuskan kunci KMS memiliki setidaknya satu alias dengan “Uji.” Tanpa `ForAnyValue` syarat, pernyataan kebijakan ini akan memungkinkan prinsipal untuk menggunakan kunci KMS yang tidak memiliki alias.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
```

```
        "alias/*Test*"
      ]
    }
  }
}
```

km: ReplicaRegion

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ReplicaRegion`	String (daftar)	Bernilai tunggal	`ReplicateKey`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk membatasi Wilayah AWS di mana prinsipal dapat mereplikasi kunci Multi-wilayah. Kunci `kms:ReplicaRegion` kondisi mengontrol akses ke ReplicateKeyoperasi berdasarkan nilai ReplicaRegionparameter dalam permintaan. Parameter ini menentukan Wilayah AWS untuk kunci replika baru.

Nilai kondisi adalah satu atau lebih Wilayah AWS nama, seperti `us-east-1` atau`ap-southeast-2`, atau pola nama, sepertieu-*. Untuk daftar nama Wilayah AWS yang AWS KMS mendukung, lihat AWS Key Management Service titik akhir dan kuota di. Referensi Umum AWS

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:ReplicaRegion` kondisi untuk mengizinkan prinsipal memanggil ReplicateKeyoperasi hanya jika nilai `ReplicaRegion` parameter adalah salah satu Wilayah yang ditentukan.

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

Kunci kondisi ini mengontrol akses hanya ke ReplicateKeyoperasi. Untuk mengontrol akses ke UpdatePrimaryRegionoperasi, gunakan kms: tombol PrimaryRegion kondisi.

km: RetiringPrincipal

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:RetiringPrincipal`	String (daftar)	Bernilai tunggal	`CreateGrant`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol kondisi ini untuk mengontrol akses ke CreateGrantoperasi berdasarkan nilai RetiringPrincipalparameter dalam permintaan. Misalnya, Anda dapat membuat hibah untuk menggunakan kunci KMS hanya jika `RetiringPrincipal` dalam `CreateGrant` permintaan cocok dengan `RetiringPrincipal` dalam pernyataan kondisi.

Untuk menentukan prinsipal pensiun, gunakan Amazon Resource Name (ARN) dari AWS prinsipal. Prinsipal yang valid meliputi Akun AWS, pengguna IAM, peran IAM, pengguna federasi, dan pengguna peran yang diasumsikan. Untuk bantuan dengan sintaks ARN untuk prinsipal, lihat ARN IAM di Panduan Pengguna IAM.

Contoh pernyataan kebijakan kunci berikut memungkinkan pengguna untuk membuat hibah untuk kunci KMS. Kunci `kms:RetiringPrincipal` kondisi membatasi izin untuk `CreateGrant` permintaan di mana kepala sekolah pensiun dalam hibah adalah. `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"
    }
  }
}
```

Lihat juga

- km: GrantConstraintType
- km: GrantIsFor AWSResource
- km: GrantOperations
- km: GranteePrincipal

km: RotationPeriodInDays

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:RotationPeriodInDays`	Numerik	Bernilai tunggal	`EnableKeyRotation`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk membatasi nilai yang dapat ditentukan oleh prinsipal dalam `RotationPeriodInDays` parameter permintaan. EnableKeyRotation

`RotationPeriodInDays`Menentukan jumlah hari antara setiap tanggal rotasi kunci otomatis. AWS KMS memungkinkan Anda untuk menentukan periode rotasi antara 90 dan 2560 hari, tetapi Anda dapat menggunakan kunci `kms:RotationPeriodInDays` kondisi untuk lebih membatasi periode rotasi, seperti menegakkan periode rotasi minimum dalam rentang yang valid.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:RotationPeriodInDays` kondisi untuk mencegah prinsipal mengaktifkan rotasi kunci jika periode rotasi kurang dari atau sama dengan 180 hari.

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

```
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:RotationPeriodInDays" : "180"
      }
  }
}
```

km: ScheduleKeyDeletionPendingWindowInDays

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ScheduleKeyDeletionPendingWindowInDays`	Numerik	Bernilai tunggal	`ScheduleKeyDeletion`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan kunci kondisi ini untuk membatasi nilai yang dapat ditentukan oleh prinsipal dalam `PendingWindowInDays` parameter permintaan. ScheduleKeyDeletion

`PendingWindowInDays`Menentukan jumlah hari yang AWS KMS akan menunggu sebelum menghapus kunci. AWS KMS memungkinkan Anda untuk menentukan masa tunggu antara 7 dan 30 hari, tetapi Anda dapat menggunakan kunci `kms:ScheduleKeyDeletionPendingWindowInDays` kondisi untuk lebih membatasi masa tunggu, seperti menegakkan periode tunggu minimum dalam rentang yang valid.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:ScheduleKeyDeletionPendingWindowInDays` kondisi untuk mencegah prinsipal menjadwalkan penghapusan kunci jika masa tunggu kurang dari atau sama dengan 21 hari.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
```

```
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
      }
  }
}
```

km: SigningAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:SigningAlgorithm`	String	Bernilai tunggal	`Sign` `Verify`	Kebijakan kunci dan kebijakan IAM

Anda dapat menggunakan tombol `kms:SigningAlgorithm` kondisi untuk mengontrol akses ke operasi Tanda dan Verifikasi berdasarkan nilai SigningAlgorithmparameter dalam permintaan. Kunci kondisi ini tidak berpengaruh pada operasi yang dilakukan di luar AWS KMS, seperti memverifikasi tanda tangan dengan kunci publik dalam key pair KMS asimetris di luar. AWS KMS

Kebijakan kunci contoh berikut memungkinkan pengguna yang dapat mengambil `testers` peran untuk menggunakan kunci KMS untuk menandatangani pesan hanya ketika algoritma penandatanganan yang digunakan untuk permintaan adalah algoritma RSASSA_PSS, seperti. `RSASSA_PSS_SHA512`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
  }
}
```

Lihat juga

- km: EncryptionAlgorithm
- the section called “km: MacAlgorithm”
- the section called “km: MessageType”

km: ValidTo

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ValidTo`	Timestamp	Bernilai tunggal	`ImportKeyMaterial`	Kebijakan kunci dan kebijakan IAM

Kunci `kms:ValidTo` kondisi mengontrol akses ke ImportKeyMaterialoperasi berdasarkan nilai ValidToparameter dalam permintaan, yang menentukan kapan bahan kunci yang diimpor kedaluwarsa. Nilainya dinyatakan dalam waktu Unix.

Secara default, parameter `ValidTo` diperlukan dalam permintaan `ImportKeyMaterial`. Namun, jika nilai ExpirationModelparameternya`KEY_MATERIAL_DOES_NOT_EXPIRE`, `ValidTo` parameternya tidak valid. Anda juga dapat menggunakan kms: ExpirationModel condition key untuk meminta `ExpirationModel` parameter atau nilai parameter tertentu.

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengimpor materi kunci ke dalam kunci KMS. Kunci syarat `kms:ValidTo` membatasi izin untuk permintaan `ImportKeyMaterial` di mana nilai `ValidTo` kurang dari atau sama dengan `1546257599.0` (31 Desember 2018 pukul 11.59.59).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
```

```
      "kms:ValidTo": "1546257599.0"
    }
  }
}
```

Lihat juga

- km: ExpirationModel
- km: WrappingAlgorithm
- km: WrappingKeySpec

km: ViaService

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:ViaService`	String	Bernilai tunggal	Operasi sumber daya utama KMS	Kebijakan kunci dan kebijakan IAM

Kunci `kms:ViaService` kondisi membatasi penggunaan kunci KMS untuk permintaan dari AWS layanan tertentu. Anda dapat menentukan satu atau beberapa layanan di setiap kunci syarat `kms:ViaService`. Operasi harus berupa operasi sumber daya kunci KMS, yaitu operasi yang diotorisasi untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam Tabel Tindakan dan Sumber Daya, cari nilai `KMS key` di `Resources` kolom untuk operasi.

Misalnya, pernyataan kebijakan kunci berikut menggunakan kunci `kms:ViaService` kondisi untuk mengizinkan kunci yang dikelola pelanggan digunakan untuk tindakan yang ditentukan hanya jika permintaan berasal dari Amazon EC2 atau Amazon RDS di wilayah AS Barat (Oregon) atas nama. `ExampleRole`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
```

```
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "ec2.us-west-2.amazonaws.com",
        "rds.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Anda juga dapat menggunakan kunci `kms:ViaService` kondisi untuk menolak izin menggunakan kunci KMS ketika permintaan berasal dari layanan tertentu. Misalnya, pernyataan kebijakan berikut dari kebijakan kunci menggunakan kunci `kms:ViaService` kondisi untuk mencegah kunci yang dikelola pelanggan digunakan untuk `Encrypt` operasi saat permintaan berasal AWS Lambda atas nama`ExampleRole`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
          "lambda.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Important

Saat Anda menggunakan kunci syarat `kms:ViaService`, layanan membuat permintaan atas nama perwakilan di Akun AWS. Perwakilan ini harus memiliki izin berikut:

- Izin untuk menggunakan kunci KMS. Prinsipal perlu memberikan izin ini ke layanan terintegrasi sehingga layanan dapat menggunakan kunci yang dikelola pelanggan atas nama kepala sekolah. Untuk informasi selengkapnya, lihat Bagaimana layanan AWS menggunakan AWS KMS.
- Izin untuk menggunakan layanan terintegrasi. Untuk detail tentang memberi pengguna akses ke AWS layanan yang terintegrasi dengan AWS KMS, lihat dokumentasi untuk layanan terintegrasi.

Semua Kunci yang dikelola AWSmenggunakan kunci `kms:ViaService` kondisi dalam dokumen kebijakan kunci mereka. Kondisi ini memungkinkan kunci KMS untuk digunakan hanya untuk permintaan yang berasal dari layanan yang membuat kunci KMS. Untuk melihat kebijakan kunci untuk sebuah Kunci yang dikelola AWS, gunakan GetKeyPolicyoperasi.

Kunci syarat `kms:ViaService` valid di IAM dan pernyataan kebijakan kunci. Layanan yang Anda tentukan harus terintegrasi dengan AWS KMS dan mendukung kunci syarat `kms:ViaService`.

Layanan yang mendukung kunci syarat **`kms:ViaService`**

Tabel berikut mencantumkan AWS layanan yang terintegrasi dengan AWS KMS dan mendukung penggunaan kunci `kms:ViaService` kondisi dalam kunci yang dikelola pelanggan Layanan dalam tabel ini mungkin tidak tersedia di semua wilayah. Gunakan `.amazonaws.com` akhiran AWS KMS ViaService nama di semua AWS partisi.

Note

Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Nama layanan	AWS KMS ViaService nama
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`
AWS Backup Gerbang	`backup-gateway.` *`AWS_region`* `.amazonaws.com`
Amazon Chime SDK	`chimevoiceconnector.` *`AWS_region`* `.amazonaws.com`
AWS CodeArtifact	`codeartifact.` *`AWS_region`* `.amazonaws.com`
CodeGuru Peninjau Amazon	`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`

Nama layanan	AWS KMS ViaService nama
Amazon Q di 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` (Hanya EBS)
Amazon Elastic Container Registry (Amazon ECR)	`ecr.`*`AWS_region`*` .amazonaws.com`
Amazon Elastic File System (Amazon EFS)	`elasticfilesystem.`*`AWS_region`*` .amazonaws.com`
Amazon ElastiCache	Sertakan kedua ViaService nama dalam nilai kunci kondisi: • `elasticache.`*`AWS_region`*` .amazonaws.com` • `dax.`*`AWS_region`*` .amazonaws.com`
AWS Elemental MediaTailor	`mediatailor.`*`AWS_region`*` .amazonaws.com`
AWS Resolusi Entitas	`entityresolution.`*`AWS_region`*` .amazonaws.com`

Nama layanan	AWS KMS ViaService nama
Amazon FinSpace	finspace. *AWS_region* .amazonaws.com
Amazon Forecast	forecast. *AWS_region* .amazonaws.com
Amazon FSx	fsx.*AWS_region* .amazonaws.com
AWS Glue	glue.*AWS_region* .amazonaws.com
AWS Ground Station	groundstation. *AWS_region* .amazonaws.com
Amazon GuardDuty	malware-protection. *AWS_region* .amazonaws.com
AWS HealthLake	healthlake. *AWS_region* .amazonaws.com
AWS IoT SiteWise	iotsitewise. *AWS_region* .amazonaws.com
Amazon Kendra	kendra.*AWS_region* .amazonaws.com
Amazon Keyspaces (untuk 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

Nama layanan	AWS KMS ViaService nama
AWS License Manager	`license-manager.` *`AWS_region`* `.amazonaws.com`
Amazon Location Service	`geo.`*`AWS_region`* `.amazonaws.com`
Amazon Lookout for Equipment	`lookoutequipment.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout for Metrics	`lookoutmetrics.` *`AWS_region`* `.amazonaws.com`
Amazon Lookout for Vision	`lookoutvision.` *`AWS_region`* `.amazonaws.com`
Amazon Macie	`macie.`*`AWS_region`* `.amazonaws.com`
AWS Mainframe Modernization	`m2.`*`AWS_region`* `.amazonaws.com`
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 for 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`

Nama layanan	AWS KMS ViaService nama
AWS HealthOmics	omics.*AWS_region* .amazonaws.com
OpenSearch Layanan Amazon	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
Wawasan Performa Amazon RDS	rds.*AWS_region* .amazonaws.com
Amazon Redshift	redshift. *AWS_region* .amazonaws.com
Editor kueri Amazon Redshift V2	sqlworkbench. *AWS_region* .amazonaws.com
Amazon Redshift Tanpa Server	redshift-serverless. *AWS_region* .amazonaws.com
Amazon Rekognition	rekognition. *AWS_region* .amazonaws.com
Amazon Relational Database Service (Amazon RDS)	rds.*AWS_region* .amazonaws.com
Toko Data Replikasi Amazon	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

Nama layanan	AWS KMS ViaService nama
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`
AWS Systems Manager Incident Manager Kontak	`ssm-contacts.`*`AWS_region`*` .amazonaws.com`
Amazon Timestream	`timestream.`*`AWS_region`*` .amazonaws.com`
Amazon Translate	`translate.`*`AWS_region`*` .amazonaws.com`
Akses Terverifikasi AWS	`verified-access.`*`AWS_region`*` .amazonaws.com`
Amazon WorkMail	`workmail.`*`AWS_region`*` .amazonaws.com`
Amazon WorkSpaces	`workspaces.`*`AWS_region`*` .amazonaws.com`
Klien WorkSpaces Tipis Amazon	`thinclient.`*`AWS_region`*` .amazonaws.com`

Nama layanan	AWS KMS ViaService nama
WorkSpaces Web Amazon	`workspaces-web.` *`AWS_region`* `.amazonaws.com`
AWS X-Ray	`xray.`*`AWS_region`*` .amazonaws.com`

km: WrappingAlgorithm

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:WrappingAlgorithm`	String	Bernilai tunggal	`GetParametersForImport`	Kebijakan kunci dan kebijakan IAM

Kunci kondisi ini mengontrol akses ke GetParametersForImportoperasi berdasarkan nilai WrappingAlgorithmparameter dalam permintaan. Anda dapat menggunakan syarat ini untuk mewajibkan perwakilan menggunakan algoritme tertentu untuk mengenkripsi materi kunci selama proses impor. Permintaan untuk kunci publik dan token impor yang diperlukan gagal ketika permintaan menentukan algoritme pembungkusan yang berbeda.

Contoh pernyataan kebijakan kunci berikut menggunakan kunci syarat `kms:WrappingAlgorithm` untuk memberikan contoh izin pengguna untuk memanggil operasi `GetParametersForImport`, tetapi mencegah mereka menggunakan algoritme pembungkusan `RSAES_OAEP_SHA_1`. Ketika `WrappingAlgorithm` dalam permintaan `GetParametersForImport` adalah `RSAES_OAEP_SHA_1`, operasi gagal.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
```

```
    }
  }
}
```

Lihat juga

- km: ExpirationModel
- km: ValidTo
- km: WrappingKeySpec

km: WrappingKeySpec

AWS KMS kunci kondisi	Jenis syarat	Jenis nilai	Operasi API	Jenis kebijakan
`kms:WrappingKeySpec`	String	Bernilai tunggal	`GetParametersForImport`	Kebijakan kunci dan kebijakan IAM

Kunci kondisi ini mengontrol akses ke GetParametersForImportoperasi berdasarkan nilai WrappingKeySpecparameter dalam permintaan. Anda dapat menggunakan syarat ini untuk mewajibkan perwakilan menggunakan jenis kunci publik tertentu selama proses impor. Jika permintaan menentukan jenis kunci yang berbeda, permintaan akan gagal.

Karena satu-satunya nilai yang valid untuk nilai parameter `WrappingKeySpec` adalah `RSA_2048`, mencegah pengguna menggunakan nilai ini akan secara efektif mencegah pengguna menggunakan operasi `GetParametersForImport`.

Contoh pernyataan kebijakan berikut menggunakan kunci syarat `kms:WrappingAlgorithm` untuk mewajibkan `WrappingKeySpec` dalam permintaan adalah `RSA_4096`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
```

```
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

Lihat juga

- km: ExpirationModel
- km: ValidTo
- km: WrappingAlgorithm

AWS KMS kunci kondisi untuk AWS Nitro Enclave

AWS Nitro Enclave adalah kemampuan Amazon EC2 yang memungkinkan Anda membuat lingkungan komputasi terisolasi yang disebut enclaves untuk melindungi dan memproses data yang sangat sensitif. AWS KMS menyediakan kunci kondisi untuk mendukung Enklaf AWS Nitro. Kunci kondisi ini hanya efektif untuk permintaan ke AWS KMS Enklave Nitro.

Saat Anda memanggil operasi Dekripsi,, GenerateDataKeyGenerateDataKeyPair, atau GenerateRandomAPI dengan dokumen pengesahan yang ditandatangani dari enklave, API ini mengenkripsi plaintext dalam respons di bawah kunci publik dari dokumen pengesahan, dan mengembalikan ciphertext alih-alih plaintext. Ciphertext ini dapat didekripsi hanya dengan menggunakan kunci pribadi di enklave. Untuk informasi selengkapnya, lihat Bagaimana Nitro Enclaves AWS menggunakan AWS KMS.

Kunci syarat berikut memungkinkan Anda membatasi izin untuk operasi ini berdasarkan konten dokumen pengesahan yang ditandatangani. Sebelum mengizinkan operasi, AWS KMS bandingkan dokumen pengesahan dari enclave dengan nilai dalam kunci kondisi ini. AWS KMS

km:RecipientAttestation: 384 ImageSha

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan
`kms:RecipientAttes`	String	Bernilai tunggal	`Decrypt`	Kebijakan kunci dan kebijakan IAM

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`tation:Im ageSha384`			`GenerateD ataKey`
			
			`GenerateD ataKeyPair`
			
			`GenerateR andom`	

Kunci `kms:RecipientAttestation:ImageSha384` kondisi mengontrol akses ke`Decrypt`,`GenerateDataKey`,`GenerateDataKeyPair`, dan `GenerateRandom` dengan kunci KMS saat gambar digest dari dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan nilai dalam kunci kondisi. `ImageSha384`Nilai sesuai dengan PCR0 dalam dokumen pengesahan. Kunci kondisi ini hanya efektif jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani untuk enclave Nitro. AWS

Nilai ini juga termasuk dalam CloudTrailacara untuk permintaan ke AWS KMS kantong Nitro.

> **Note**
>
> Kunci kondisi ini valid dalam pernyataan kebijakan utama dan pernyataan kebijakan IAM meskipun tidak muncul di konsol IAM atau Referensi Otorisasi Layanan IAM.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran untuk menggunakan kunci KMS untuk Dekripsi,, GenerateDataKeyGenerateDataKeyPair, dan operasi. GenerateRandom Kunci `kms:RecipientAttestation:ImageSha384` kondisi memungkinkan operasi hanya jika nilai intisari gambar (PCR0) dari dokumen pengesahan dalam permintaan cocok dengan nilai intisari gambar dalam kondisi. Kunci kondisi ini hanya efektif jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani untuk enclave Nitro. AWS

Jika permintaan tidak menyertakan dokumen pengesahan yang valid dari kantong AWS Nitro, izin ditolak karena kondisi ini tidak terpenuhi.

```
{
```

```
  "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
    }
  }
}
```

km ::PCR RecipientAttestation <PCR_ID>

AWS KMS Kunci Kondisi	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:PCR<PCR_ID>`	String	Bernilai tunggal	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Kebijakan kunci dan kebijakan IAM

Kunci `kms:RecipientAttestation:PCR<PCR_ID>` kondisi mengontrol akses ke`Decrypt`,, `GenerateDataKeyGenerateDataKeyPair`, dan `GenerateRandom` dengan kunci KMS hanya jika konfigurasi platform register (PCR) dari dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan PCR dalam kunci kondisi. Kunci kondisi ini hanya efektif jika `Recipient`

parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani dari enklave Nitro. AWS

Nilai ini juga termasuk dalam CloudTrailperistiwa yang mewakili permintaan AWS KMS untuk kantong Nitro.

> **Note**
>
> Kunci kondisi ini valid dalam pernyataan kebijakan utama dan pernyataan kebijakan IAM meskipun tidak muncul di konsol IAM atau Referensi Otorisasi Layanan IAM.

Untuk menentukan nilai PCR, gunakan format berikut. Gabungkan ID PCR ke nama kunci syarat. Nilai PCR harus berupa string heksadesimal huruf kecil hingga 96 byte.

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

Misalnya, kunci kondisi berikut menentukan nilai tertentu untuk PCR1, yang sesuai dengan hash kernel yang digunakan untuk enclave dan proses bootstrap.

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

Contoh pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran untuk menggunakan kunci KMS untuk operasi Dekripsi.

Kunci syarat `kms:RecipientAttestation:PCR` dalam pernyataan ini memungkinkan operasi hanya jika nilai PCR1 dalam dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan nilai `kms:RecipientAttestation:PCR1` dalam syarat. Gunakan operator kebijakan `StringEqualsIgnoreCase` untuk mewajibkan perbandingan nilai PCR yang tidak peka huruf besar/kecil.

Jika permintaan tidak menyertakan dokumen pengesahan, izin ditolak karena kondisi ini tidak terpenuhi.

```
{
  "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 untuk AWS KMS

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. AWS KMSmendukung ABAC dengan memungkinkan Anda untuk mengontrol akses ke kunci yang dikelola pelanggan Anda berdasarkan tag dan alias yang terkait dengan kunci KMS. Kunci kondisi tag dan alias yang memungkinkan ABAC AWS KMS menyediakan cara yang kuat dan fleksibel untuk mengotorisasi prinsipal untuk menggunakan kunci KMS tanpa mengedit kebijakan atau mengelola hibah. Namun Anda harus menggunakan fitur ini dengan hati-hati sehingga perwakilan tidak ditolak aksesnya atau diizinkan secara tidak sengaja.

Jika Anda menggunakan ABAC, ketahui bahwa izin untuk mengelola tag dan alias sekarang adalah izin kontrol akses. Pastikan Anda mengetahui tag dan alias yang ada di semua kunci KMS sebelum menerapkan kebijakan yang bergantung pada tag atau alias. Lakukan tindakan pencegahan yang wajar saat menambahkan, menghapus, dan memperbarui alias, dan saat menandai dan menghapus tanda kunci. Berikan izin untuk mengelola tag dan alias hanya kepada perwakilan yang membutuhkannya, dan membatasi tag dan alias yang dapat mereka kelola.

> **Catatan**
>
> Saat menggunakan ABAC untuk AWS KMS, hati-hati saat memberikan izin kepada perwakilan untuk mengelola tag dan alias. Mengubah tag atau alias mungkin mengizinkan atau menolak izin ke kunci KMS. Administrator kunci yang tidak memiliki izin untuk mengubah kebijakan kunci atau membuat hibah dapat mengontrol akses ke kunci KMS jika mereka memiliki izin untuk mengelola tag atau alias.
> Mungkin diperlukan waktu hingga lima menit untuk perubahan tag dan alias untuk memengaruhi otorisasi kunci KMS. Perubahan terbaru mungkin terlihat dalam operasi API sebelum mempengaruhi otorisasi.

Untuk mengontrol akses ke kunci KMS berdasarkan aliasnya, Anda harus menggunakan tombol kondisi. Anda tidak dapat menggunakan alias untuk mewakili kunci KMS dalam `Resource` elemen pernyataan kebijakan. Ketika alias muncul di `Resource` elemen, pernyataan kebijakan berlaku untuk alias, bukan ke kunci KMS terkait.

Pelajari selengkapnya

- Untuk detail tentang dukungan AWS KMS untuk ABAC, termasuk contoh, lihat Menggunakan alias untuk mengontrol akses ke tombol KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.
- Untuk informasi umum selengkapnya tentang penggunaan tag untuk mengontrol akses ke sumber daya AWS, lihat Apa guna ABAC AWS? dan Mengontrol Akses ke Sumber Daya AWS Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

Kunci syarat ABAC untuk AWS KMS

Untuk mengotorisasi akses ke kunci KMS berdasarkan tag dan aliasnya, gunakan kunci kondisi berikut dalam kebijakan kunci atau kebijakan IAM.

Kunci syarat ABAC	Deskripsi	Jenis kebijakan	Operasi AWS KMS
aws: ResourceTag	Tag (kunci dan nilai) pada kunci KMS cocok dengan tag (kunci dan nilai) atau pola tag dalam kebijakan	Khusus kebijakan IAM	Operasi sumber daya utama KMS 2
aws:RequestTag/tag-kunci	Tag (kunci dan nilai) dalam permintaan cocok dengan tag (kunci dan nilai) atau pola tag dalam kebijakan	Kebijakan kunci dan kebijakan IAM[1]	TagResource, UntagResource

Kunci syarat ABAC	Deskripsi	Jenis kebijakan	Operasi AWS KMS
aws: TagKeys	Kunci tag dalam permintaan cocok dengan kunci tag dalam kebijakan	Kebijakan kunci dan kebijakan IAM[1]	TagResource, UntagResource
km: ResourceAliases	Alias yang terkait dengan kunci KMS cocok dengan alias atau pola alias dalam kebijakan	Khusus kebijakan IAM	Operasi sumber daya utama KMS 2
km: RequestAlias	Alias yang mewakili kunci KMS dalam permintaan cocok dengan pola alias atau alias dalam kebijakan.	Kebijakan kunci dan kebijakan IAM[1]	Operasi kriptografi,, DescribeKeyGetPublicKey

[1]Kunci syarat yang dapat digunakan dalam kebijakan kunci juga dapat digunakan dalam kebijakan IAM, tetapi hanya jika kebijakan kunci mengizinkannya.

[2] Operasi sumber daya kunci KMS adalah operasi yang diizinkan untuk kunci KMS tertentu. Untuk mengidentifikasi operasi sumber daya kunci KMS, dalam tabel AWS KMS izin, cari nilai kunci KMS di `Resources` kolom untuk operasi.

Misalnya, Anda dapat menggunakan kunci syarat ini untuk membuat kebijakan berikut.

- Kebijakan IAM dengan `kms:ResourceAliases` itu memungkinkan izin untuk menggunakan kunci KMS dengan alias atau pola alias tertentu. Ini sedikit berbeda dengan kebijakan yang bergantung pada tag: Meskipun Anda dapat menggunakan pola alias dalam kebijakan, setiap alias harus unik dalam Akun AWS dan Wilayah. Ini memungkinkan Anda menerapkan kebijakan ke kumpulan kunci KMS tertentu tanpa mencantumkan ARN kunci kunci kunci KMS dalam pernyataan kebijakan. Untuk menambah atau menghapus kunci KMS dari set, ubah alias tombol KMS.

- Kebijakan kunci dengan `kms:RequestAlias` itu memungkinkan prinsipal untuk menggunakan kunci KMS dalam `Encrypt` operasi, tetapi hanya ketika `Encrypt` permintaan menggunakan alias tersebut untuk mengidentifikasi kunci KMS.
- Kebijakan IAM dengan `aws:ResourceTag/`*`tag-key`* itu menolak izin untuk menggunakan kunci KMS dengan kunci tag dan nilai tag tertentu. Ini memungkinkan Anda menerapkan kebijakan ke kumpulan kunci KMS tertentu tanpa mencantumkan ARN kunci kunci kunci KMS dalam pernyataan kebijakan. Untuk menambah atau menghapus kunci KMS dari set, tag atau untag kunci KMS.
- Kebijakan IAM dengan `aws:RequestTag/`*`tag-key`* itu memungkinkan prinsipal untuk menghapus hanya `"Purpose"="Test"` tag kunci KMS.
- Kebijakan IAM dengan `aws:TagKeys` itu menolak izin untuk menandai atau menghapus tag kunci KMS dengan kunci tag. `Restricted`

ABAC menjadikan manajemen akses fleksibel dan dapat diskalakan. Misalnya, Anda dapat menggunakan kunci `aws:ResourceTag/`*`tag-key`* kondisi untuk membuat kebijakan IAM yang memungkinkan prinsipal menggunakan kunci KMS untuk operasi tertentu hanya jika kunci KMS memiliki tag. `Purpose=Test` Kebijakan ini berlaku untuk semua kunci KMS di seluruh Wilayah. Akun AWS

Saat dilampirkan ke pengguna atau peran, kebijakan IAM berikut memungkinkan prinsipal untuk menggunakan semua kunci KMS yang ada dengan `Purpose=Test` tag untuk operasi yang ditentukan. Untuk menyediakan akses ini ke kunci KMS baru atau yang sudah ada, Anda tidak perlu mengubah kebijakan. Cukup lampirkan `Purpose=Test` tag ke tombol KMS. Demikian pula, untuk menghapus akses ini dari kunci KMS dengan `Purpose=Test` tag, edit atau hapus 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"
        }
      }
    }
  ]
}
```

Namun, jika Anda menggunakan fitur ini, hati-hati saat mengelola tag dan alias. Menambahkan, mengubah, atau menghapus tag atau alias dapat secara tidak sengaja mengizinkan atau menolak akses ke kunci KMS. Administrator kunci yang tidak memiliki izin untuk mengubah kebijakan kunci atau membuat hibah dapat mengontrol akses ke kunci KMS jika mereka memiliki izin untuk mengelola tag dan alias. Untuk mengurangi risiko ini, pertimbangkan membatasi izin untuk mengelola tag dan alias. Misalnya, Anda mungkin ingin hanya mengizinkan perwakilan terpilih yang dapat mengelola tag `Purpose=Test`. Untuk detailnya, lihat Menggunakan alias untuk mengontrol akses ke tombol KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.

Tag atau alias?

AWS KMS mendukung ABAC dengan tag dan alias. Kedua opsi ini menyediakan strategi kontrol akses yang fleksibel dan dapat diskalakan, tetapi keduanya sedikit berbeda satu sama lain.

Anda mungkin memutuskan untuk menggunakan tag atau menggunakan alias berdasarkan pola penggunaan AWS tertentu. Misalnya, jika Anda telah memberikan izin penandaan kepada sebagian besar administrator, mungkin akan lebih mudah untuk mengontrol strategi otorisasi berdasarkan alias. Atau, jika Anda mendekati kuota alias per kunci KMS, Anda mungkin lebih memilih strategi otorisasi berdasarkan tag.

Manfaat berikut adalah kepentingan umum.

Manfaat kontrol akses berbasis tag

- Mekanisme otorisasi yang sama untuk jenis sumber daya AWS yang berbeda.

  Anda dapat menggunakan tag atau kunci tag yang sama untuk mengontrol akses ke beberapa jenis sumber daya, seperti klaster Amazon Relational Database Service (Amazon RDS), volume Amazon Elastic Block Store (Amazon EBS) Block Store (Amazon EBS), dan kunci KMS. Fitur ini mengaktifkan beberapa model otorisasi yang berbeda yang lebih fleksibel dari kontrol akses berbasis peran tradisional.
- Otorisasi akses ke sekelompok kunci KMS.

Anda dapat menggunakan tag untuk mengelola akses ke sekelompok kunci KMS yang sama Akun AWS dan Wilayah. Tetapkan tag atau kunci tag yang sama ke kunci KMS yang Anda pilih. Kemudian buat pernyataan easy-to-maintain kebijakan sederhana yang didasarkan pada tag atau kunci tag. Untuk menambah atau menghapus kunci KMS dari grup otorisasi Anda, tambahkan atau hapus tag; Anda tidak perlu mengedit kebijakan.

Manfaat kontrol akses berbasis alias

- Otorisasi akses ke operasi kriptografi berdasarkan alias.

  Sebagian besar kondisi kebijakan berbasis permintaan untuk atribut, termasuk aws:RequestTag/tag-key, hanya memengaruhi operasi yang menambahkan, mengedit, atau menghapus atribut. Tetapi kms: RequestAlias condition key mengontrol akses ke operasi kriptografi berdasarkan alias yang digunakan untuk mengidentifikasi kunci KMS dalam permintaan. Misalnya, Anda dapat memberikan izin utama untuk menggunakan kunci KMS dalam `Encrypt` operasi tetapi hanya jika nilai `KeyId` parameternya`alias/restricted-key-1`. Untuk memenuhi syarat ini, Anda memerlukan semua hal berikut:

  - Kunci KMS harus dikaitkan dengan alias itu.
  - Permintaan harus menggunakan alias untuk mengidentifikasi kunci KMS.
  - Kepala sekolah harus memiliki izin untuk menggunakan kunci KMS sesuai dengan `kms:RequestAlias` kondisi tersebut.

  Ini sangat berguna jika aplikasi Anda biasanya menggunakan nama alias atau alias ARN untuk merujuk ke kunci KMS.
- Berikan izin yang sangat terbatas.

  Alias harus unik dalam Akun AWS dan Wilayah. Akibatnya, memberi prinsipal akses ke kunci KMS berdasarkan alias bisa jauh lebih membatasi daripada memberi mereka akses berdasarkan tag. Tidak seperti alias, tag dapat ditetapkan ke beberapa kunci KMS di akun dan Wilayah yang sama. Jika Anda memilih, Anda dapat menggunakan pola alias, seperti`alias/test*`, untuk memberikan akses prinsipal ke sekelompok kunci KMS di akun dan Wilayah yang sama. Namun, mengizinkan atau menolak akses ke alias tertentu memungkinkan kontrol yang sangat ketat pada kunci KMS.

Memecahkan Masalah ABAC untuk AWS KMS

Mengontrol akses ke kunci KMS berdasarkan tag dan aliasnya nyaman dan kuat. Namun, ini rentan akan beberapa kesalahan yang dapat diprediksi yang ingin Anda cegah.

Akses berubah karena perubahan tag

Jika tag dihapus atau nilainya diubah, prinsipal yang memiliki akses ke kunci KMS hanya berdasarkan tag tersebut akan ditolak akses ke kunci KMS. Ini juga dapat terjadi ketika tag yang disertakan dalam pernyataan kebijakan penolakan ditambahkan ke kunci KMS. Menambahkan tag terkait kebijakan ke kunci KMS dapat memungkinkan akses ke kepala sekolah yang harus ditolak aksesnya ke kunci KMS.

Misalnya, misalkan kepala sekolah memiliki akses ke kunci KMS berdasarkan `Project=Alpha` tag, seperti izin yang diberikan oleh contoh pernyataan kebijakan IAM berikut.

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

Jika tag dihapus dari kunci KMS atau nilai tag diubah, prinsipal tidak lagi memiliki izin untuk menggunakan kunci KMS untuk operasi yang ditentukan. Ini mungkin menjadi jelas ketika prinsipal mencoba membaca atau menulis data dalam AWS layanan yang menggunakan kunci yang dikelola pelanggan Untuk melacak perubahan tag, tinjau CloudTrail log TagResourceatau UntagResource entri Anda.

Untuk memulihkan akses tanpa memperbarui kebijakan, ubah tag pada tombol KMS. Tindakan ini memiliki dampak minimal selain jangka waktu singkat saat diterapkan di seluruh AWS KMS. Agar kesalahan seperti ini tidak terjadi, hanya berikan izin penandaan dan penghapusan tanda untuk perwakilan yang memerlukannya dan batasi izin penandaan untuk tag yang perlu dikelola.. Sebelum mengubah tag, cari kebijakan untuk mendeteksi akses yang bergantung pada tag, dan dapatkan kunci KMS di semua Wilayah yang memiliki tag. Anda dapat mempertimbangkan untuk membuat CloudWatch alarm Amazon ketika tag tertentu diubah.

Perubahan akses karena perubahan alias

Jika alias dihapus atau dikaitkan dengan kunci KMS yang berbeda, prinsipal yang memiliki akses ke kunci KMS hanya berdasarkan alias tersebut akan ditolak akses ke kunci KMS. Hal ini juga dapat terjadi ketika alias yang terkait dengan kunci KMS disertakan dalam pernyataan kebijakan penolakan. Menambahkan alias terkait kebijakan ke kunci KMS juga dapat memungkinkan akses ke kepala sekolah yang harus ditolak aksesnya ke kunci KMS.

Misalnya, pernyataan kebijakan IAM berikut menggunakan kms: ResourceAliases condition key untuk mengizinkan akses ke kunci KMS di Wilayah akun yang berbeda dengan alias yang ditentukan.

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

```
  ]
}
```

Untuk melacak perubahan alias, tinjau CloudTrail log Anda untuk CreateAlias, UpdateAlias, dan DeleteAliasentri.

Untuk memulihkan akses tanpa memperbarui kebijakan, ubah alias yang terkait dengan kunci KMS. Karena setiap alias dapat dikaitkan dengan hanya satu kunci KMS di akun dan Wilayah, mengelola alias sedikit lebih sulit daripada mengelola tag. Memulihkan akses ke beberapa prinsipal pada satu kunci KMS dapat menolak akses prinsipal yang sama atau lainnya ke kunci KMS yang berbeda.

Agar kesalahan ini tidak terjadi, berikan izin manajemen alias hanya untuk perwakilan yang memerlukannya dan batasi izin manajemen-alias untuk alias yang perlu dikelola. Sebelum memperbarui atau menghapus alias, cari kebijakan untuk mendeteksi akses yang bergantung pada alias, dan temukan kunci KMS di semua Wilayah yang terkait dengan alias.

Akses ditolak karena kuota alias

Pengguna yang diberi wewenang untuk menggunakan kunci KMS dengan ResourceAliases kondisi kms: akan mendapatkan `AccessDenied` pengecualian jika kunci KMS melebihi alias default per kuota kunci KMS untuk akun dan Wilayah tersebut.

Untuk memulihkan akses, hapus alias yang terkait dengan kunci KMS sehingga sesuai dengan kuota. Atau gunakan mekanisme alternatif untuk memberi pengguna akses ke kunci KMS.

Perubahan otorisasi yang tertunda

Perubahan yang Anda buat pada tag dan alias mungkin membutuhkan waktu hingga lima menit untuk memengaruhi otorisasi kunci KMS. Akibatnya, perubahan tag atau alias mungkin tercermin dalam respons dari operasi API sebelum berlaku pada otorisasi. Penundaan ini mungkin lebih lama dari penundaan eventual consistency singkat yang berlaku pada sebagian besar operasi AWS KMS.

Misalnya, Anda mungkin memiliki kebijakan IAM yang memungkinkan prinsipal tertentu untuk menggunakan kunci KMS apa pun dengan tag. `"Purpose"="Test"` Kemudian Anda menambahkan `"Purpose"="Test"` tag ke kunci KMS. Meskipun TagResourceoperasi selesai dan ListResourceTagsrespons mengonfirmasi bahwa tag ditetapkan ke kunci KMS, kepala sekolah mungkin tidak memiliki akses ke kunci KMS hingga lima menit.

Agar tidak terjadi kesalahan, buat perkiraan penundaan ke dalam kode Anda.

Permintaan gagal karena pembaruan alias

Saat memperbarui alias, Anda mengaitkan alias yang ada dengan kunci KMS yang berbeda.

Dekripsi dan ReEncryptpermintaan yang menentukan nama alias atau alias ARN mungkin gagal karena alias sekarang dikaitkan dengan kunci KMS yang tidak mengenkripsi ciphertext. Situasi ini biasanya menampilkan `IncorrectKeyException` atau `NotFoundException`. Atau jika permintaan tidak memiliki `KeyId` atau `DestinationKeyId` parameter, operasi mungkin gagal dengan `AccessDenied` pengecualian karena pemanggil tidak lagi memiliki akses ke kunci KMS yang mengenkripsi ciphertext.

Anda dapat melacak perubahan dengan melihat CloudTrail log untuk CreateAlias, UpdateAlias, dan entri DeleteAliaslog. Anda juga dapat menggunakan nilai `LastUpdatedDate` bidang dalam ListAliasesrespons untuk mendeteksi perubahan.

Misalnya, ListAliasescontoh respons berikut menunjukkan bahwa `ProjectAlpha_Test` alias dalam `kms:ResourceAliases` kondisi telah diperbarui. Akibatnya, kepala sekolah yang memiliki akses berdasarkan alias kehilangan akses ke kunci KMS yang sebelumnya terkait. Sebaliknya, mereka memiliki akses ke kunci KMS yang baru terkait.

```
$ 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
        }
    ]
}
```

Memperbaiki perubahan ini tidaklah mudah. Anda dapat memperbarui alias lagi untuk mengaitkannya dengan kunci KMS asli. Namun, sebelum Anda bertindak, Anda perlu mempertimbangkan efek perubahan itu pada kunci KMS yang saat ini terkait. Jika kepala sekolah menggunakan kunci KMS terakhir dalam operasi kriptografi, mereka mungkin memerlukan akses lanjutan ke sana. Dalam hal ini, Anda mungkin ingin memperbarui kebijakan untuk memastikan bahwa kepala sekolah memiliki izin untuk menggunakan kedua kunci KMS.

Anda dapat mencegah kesalahan seperti ini: Sebelum memperbarui alias, cari kebijakan untuk mendeteksi akses yang bergantung pada alias. Kemudian dapatkan kunci KMS di semua Wilayah yang terkait dengan alias. Berikan izin manajemen alias hanya untuk perwakilan yang memerlukannya dan batasi izin manajemen-alias untuk alias yang perlu dikelola.

Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS

Anda dapat mengizinkan pengguna atau peran yang berbeda Akun AWS untuk menggunakan kunci KMS di akun Anda. Akses lintas akun memerlukan izin dalam kebijakan kunci kunci KMS dan dalam kebijakan IAM di akun pengguna eksternal.

Izin lintas akun hanya efektif untuk operasi berikut:

- Operasi kriptografi
- CreateGrant
- DescribeKey
- GetKeyRotationStatus
- GetPublicKey
- ListGrants
- RetireGrant
- RevokeGrant

Jika Anda memberikan izin akun yang berbeda untuk operasi lain kepada pengguna, izin tersebut tidak berpengaruh. Misalnya, jika Anda memberikan izin prinsipal di akun lain kms: ListKeys izin dalam kebijakan IAM, atau ScheduleKeyDeletion izin kms: pada kunci KMS dalam kebijakan kunci, upaya pengguna untuk memanggil operasi tersebut pada sumber daya Anda masih gagal.

Untuk detail tentang penggunaan kunci KMS di akun yang berbeda untuk AWS KMS operasi, lihat kolom Penggunaan lintas akun di danAWS KMS izin. Menggunakan kunci KMS di akun lain Ada bagian Penggunaan lintas akun juga dalam setiap deskripsi API di Referensi API AWS Key Management Service.

> **Warning**
>
> Berhati-hatilah dalam memberikan izin kepada kepala sekolah untuk menggunakan kunci KMS Anda. Sebisa mungkin, ikuti prinsip hak istimewa paling rendah. Berikan pengguna akses hanya ke kunci KMS yang mereka butuhkan hanya untuk operasi yang mereka butuhkan.
>
> Juga, berhati-hatilah dalam menggunakan kunci KMS yang tidak dikenal, terutama kunci KMS di akun yang berbeda. Pengguna jahat mungkin memberi Anda izin untuk menggunakan kunci KMS mereka untuk mendapatkan informasi tentang Anda atau akun Anda.
>
> Untuk informasi tentang menggunakan kebijakan untuk melindungi sumber daya di akun Anda, lihat Praktik terbaik untuk kebijakan IAM.

Untuk memberikan izin menggunakan kunci KMS kepada pengguna dan peran di akun lain, Anda harus menggunakan dua jenis kebijakan yang berbeda:

- Kebijakan kunci untuk kunci KMS harus memberikan izin kepada akun eksternal (atau pengguna dan peran di akun eksternal) untuk menggunakan kunci KMS. Kebijakan utama ada di akun yang memiliki kunci KMS.
- Kebijakan IAM di akun eksternal harus mendelegasikan izin kebijakan kunci untuk pengguna dan perannya. Kebijakan ini ditetapkan di akun eksternal dan akan memberikan izin kepada pengguna dan peran dalam akun tersebut.

Kebijakan kunci menentukan siapa yang dapat memiliki akses ke kunci KMS. Kebijakan IAM menentukan siapa yang memiliki akses ke kunci KMS. Kebijakan kunci atau kebijakan IAM saja tidak cukup—Anda harus mengubah keduanya.

Untuk mengedit kebijakan utama, Anda dapat menggunakan Tampilan Kebijakan di AWS Management Console atau menggunakan PutKeyPolicyoperasi CreateKeyatau. Untuk bantuan menyetel kebijakan kunci saat membuat kunci KMS, lihatMembuat kunci KMS yang dapat digunakan akun lain.

Untuk mendapatkan bantuan saat mengedit kebijakan IAM, lihat Menggunakan kebijakan IAM dengan AWS KMS.

Untuk contoh yang menunjukkan bagaimana kebijakan utama dan kebijakan IAM bekerja sama untuk mengizinkan penggunaan kunci KMS di akun yang berbeda, lihat. Contoh 2: Pengguna mengasumsikan peran dengan izin untuk menggunakan kunci KMS dalam yang berbeda Akun AWS

Anda dapat melihat AWS KMS operasi lintas akun yang dihasilkan pada kunci KMS di log AndaAWS CloudTrail. Operasi yang menggunakan kunci KMS di akun lain dicatat di akun penelepon dan akun pemilik kunci KMS.

Topik

- Langkah 1: Menambahkan pernyataan kebijakan kunci di akun lokal
- Langkah 2: Menambahkan kebijakan IAM di akun eksternal
- Membuat kunci KMS yang dapat digunakan akun lain
- Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS
- Menggunakan kunci KMS di akun lain

> **Note**
>
> Contoh dalam topik ini menunjukkan cara menggunakan kebijakan kunci dan kebijakan IAM bersama-sama untuk menyediakan dan membatasi akses ke kunci KMS. Contoh generik ini tidak dimaksudkan untuk mewakili izin yang Layanan AWS diperlukan tertentu pada kunci KMS. Untuk informasi tentang izin yang Layanan AWS diperlukan, lihat topik enkripsi dalam dokumentasi layanan.

Langkah 1: Menambahkan pernyataan kebijakan kunci di akun lokal

Kebijakan kunci untuk kunci KMS adalah penentu utama siapa yang dapat mengakses kunci KMS dan operasi mana yang dapat mereka lakukan. Kebijakan utama selalu ada di akun yang memiliki kunci KMS. Tidak seperti kebijakan IAM, kebijakan kunci tidak menentukan sumber daya. Sumber daya adalah kunci KMS yang terkait dengan kebijakan kunci. Saat memberikan izin lintas akun, kebijakan kunci untuk kunci KMS harus memberikan izin kepada akun eksternal (atau pengguna dan peran di akun eksternal) untuk menggunakan kunci KMS.

Untuk memberikan izin akun eksternal untuk menggunakan kunci KMS, tambahkan pernyataan ke kebijakan kunci yang menentukan akun eksternal. Di elemen `Principal` kebijakan kunci, masukkan Amazon Resource Name (ARN) dari akun eksternal.

Saat Anda menentukan akun eksternal dalam kebijakan kunci, administrator IAM di akun eksternal dapat menggunakan kebijakan IAM untuk mendelegasikan izin tersebut untuk setiap pengguna dan peran dalam akun eksternal. Mereka juga dapat memutuskan tindakan yang ditentukan dalam kebijakan kunci yang dapat dilakukan pengguna dan peran.

Izin yang diberikan ke akun eksternal dan prinsipalnya hanya berlaku jika akun eksternal diaktifkan di Wilayah yang menghosting kunci KMS dan kebijakan utamanya. Untuk informasi tentang Wilayah yang tidak diaktifkan secara default (“opt-in Regions”), lihat Mengelola Wilayah AWS di Referensi Umum AWS.

Misalnya, Anda ingin mengizinkan akun `444455556666` menggunakan kunci KMS enkripsi simetris di akun. `111122223333` Untuk melakukannya, tambahkan pernyataan kebijakan seperti yang ada di contoh berikut ke kebijakan kunci untuk kunci KMS di akun`111122223333`. Pernyataan kebijakan ini memberikan akun eksternal,`444455556666`, izin untuk menggunakan kunci KMS dalam operasi kriptografi untuk kunci KMS enkripsi simetris.

> **Note**
>
> Contoh berikut mewakili kebijakan kunci sampel untuk berbagi kunci KMS dengan akun lain. Ganti contoh`Sid`,`Principal`, dan `Action` nilai dengan nilai yang valid untuk tujuan penggunaan kunci KMS Anda.

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

Alih-alih memberikan izin untuk akun eksternal, Anda dapat menentukan pengguna dan peran eksternal tertentu dalam kebijakan kunci. Namun, pengguna dan peran tersebut tidak dapat menggunakan kunci KMS sampai administrator IAM di akun eksternal melampirkan kebijakan IAM yang tepat ke identitas mereka. Kebijakan IAM dapat memberikan izin untuk semua atau sebagian peran dan pengguna eksternal yang ditentukan dalam kebijakan kunci. Mereka juga dapat mengizinkan semua atau sebagian tindakan yang ditentukan dalam kebijakan kunci.

Menentukan identitas dalam kebijakan kunci akan membatasi izin yang bisa diberikan administrator IAM di akun eksternal. Namun, hal ini akan membuat manajemen kebijakan dengan dua akun menjadi lebih kompleks. Misalnya, anggaplah Anda perlu menambahkan pengguna atau peran. Anda harus menambahkan identitas tersebut ke kebijakan kunci di akun yang memiliki kunci KMS dan membuat kebijakan IAM di akun identitas.

Untuk menentukan peran atau pengguna eksternal tertentu dalam kebijakan kunci, di elemen `Principal`, masukkan Amazon Resource Name (ARN) dari pengguna atau peran di akun eksternal.

Misalnya, contoh pernyataan kebijakan kunci berikut memungkinkan `ExampleRole` di akun `444455556666` untuk menggunakan kunci KMS di akun`111122223333`. Pernyataan kebijakan kunci ini memberikan akun eksternal,`444455556666`, izin untuk menggunakan kunci KMS dalam operasi kriptografi untuk kunci KMS enkripsi simetris.

> **Note**
>
> Contoh berikut mewakili kebijakan kunci sampel untuk berbagi kunci KMS dengan akun lain. Ganti contoh`Sid`,`Principal`, dan `Action` nilai dengan nilai yang valid untuk tujuan penggunaan kunci KMS Anda.

```
{
    "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**
>
> Jangan menyetel Principal ke tanda bintang (*) dalam pernyataan kebijakan kunci apa pun yang mengizinkan izin kecuali Anda menggunakan kondisi untuk membatasi kebijakan utama. Tanda bintang memberikan setiap identitas di setiap Akun AWS izin untuk menggunakan kunci KMS, kecuali pernyataan kebijakan lain secara eksplisit menyangkalnya. Pengguna lain Akun AWS dapat menggunakan kunci KMS Anda setiap kali mereka memiliki izin yang sesuai di akun mereka sendiri.

Anda juga harus memutuskan izin yang ingin diberikan ke akun eksternal. Untuk daftar izin pada kunci KMS, lihat. AWS KMS izin

Anda dapat memberikan izin akun eksternal untuk menggunakan kunci KMS dalam operasi kriptografi dan menggunakan kunci KMS dengan AWS layanan yang terintegrasi dengannya. AWS KMS Untuk melakukannya, gunakan bagian Pengguna Kunci dari AWS Management Console. Untuk detail selengkapnya, lihat Membuat kunci KMS yang dapat digunakan akun lain.

Untuk menentukan izin lain dalam kebijakan kunci, edit dokumen kebijakan kunci. Misalnya, Anda mungkin ingin memberi pengguna izin untuk mendekripsi tetapi tidak mengenkripsi, atau izin untuk melihat kunci KMS tetapi tidak menggunakannya. Untuk mengedit dokumen kebijakan utama, Anda dapat menggunakan Tampilan Kebijakan di AWS Management Console atau CreateKeyatau PutKeyPolicyoperasi.

Langkah 2: Menambahkan kebijakan IAM di akun eksternal

Kebijakan kunci di akun yang memiliki kunci KMS menetapkan rentang izin yang valid. Namun, pengguna dan peran di akun eksternal tidak dapat menggunakan kunci KMS sampai Anda melampirkan kebijakan IAM yang mendelegasikan izin tersebut, atau menggunakan hibah untuk mengelola akses ke kunci KMS. Kebijakan IAM ditetapkan di akun eksternal.

Jika kebijakan kunci memberikan izin untuk akun eksternal, Anda dapat melampirkan kebijakan IAM untuk setiap pengguna atau peran dalam akun tersebut. Namun jika kebijakan kunci memberikan izin untuk pengguna atau peran tertentu, kebijakan IAM hanya dapat memberikan izin tersebut untuk semua atau sebagian pengguna dan peran tertentu. Jika kebijakan IAM memberikan akses kunci KMS ke pengguna atau peran eksternal lainnya, kebijakan tersebut tidak berpengaruh.

Kebijakan kunci juga membatasi tindakan dalam kebijakan IAM. Kebijakan IAM dapat mendelegasikan semua atau sebagian tindakan yang ditentukan dalam kebijakan kunci. Jika kebijakan IAM mencantumkan tindakan yang tidak ditentukan dalam kebijakan kunci, izin tersebut tidak efektif.

Contoh berikut kebijakan IAM memungkinkan prinsipal untuk menggunakan kunci KMS dalam akun `111122223333` untuk operasi kriptografi. Untuk memberikan izin ini kepada pengguna dan peran dalam akun `444455556666`, lampirkan kebijakan untuk pengguna atau peran dalam akun `444455556666`.

> **Note**
>
> Contoh berikut merupakan contoh kebijakan IAM untuk berbagi kunci KMS dengan akun lain. Ganti contoh`Sid`,`Resource`, dan `Action` nilai dengan nilai yang valid untuk tujuan penggunaan kunci KMS Anda.

```
{
    "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"
}
```

Perhatikan detail penting berikut tentang kebijakan ini:

- Tidak seperti kebijakan kunci, pernyataan kebijakan IAM tidak berisi elemen `Principal`. Dalam kebijakan IAM, perwakilan adalah identitas yang dilampirkan kebijakan tersebut.
- `Resource`Elemen dalam kebijakan IAM mengidentifikasi kunci KMS yang dapat digunakan oleh prinsipal. Untuk menentukan kunci KMS, tambahkan kunci ARN ke `Resource` elemen.
- Anda dapat menentukan lebih dari satu kunci KMS dalam `Resource` elemen. Tetapi jika Anda tidak menentukan kunci KMS tertentu dalam `Resource` elemen, Anda mungkin secara tidak sengaja memberikan akses ke lebih banyak kunci KMS daripada yang Anda inginkan.
- Untuk mengizinkan pengguna eksternal menggunakan kunci KMS dengan AWSlayanan yang terintegrasi dengannyaAWS KMS, Anda mungkin perlu menambahkan izin ke kebijakan kunci atau kebijakan IAM. Untuk detail selengkapnya, lihat Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS.

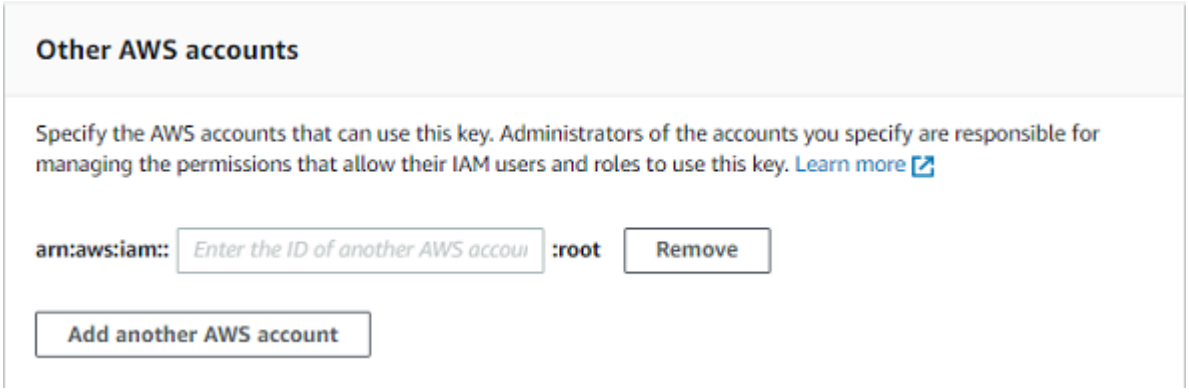
Untuk informasi selengkapnya tentang bekerja dengan kebijakan IAM, lihat Kebijakan IAM.

Membuat kunci KMS yang dapat digunakan akun lain

Bila Anda menggunakan CreateKeyoperasi untuk membuat kunci KMS, Anda dapat menggunakan `Policy` parameternya untuk menentukan kebijakan kunci yang memberikan akun eksternal, atau pengguna eksternal dan peran, izin untuk menggunakan kunci KMS. Anda juga harus menambahkan Kebijakan IAM di akun eksternal yang mendelegasikan izin ini untuk akun pengguna dan peran, bahkan jika pengguna dan peran sudah ditentukan dalam kebijakan kunci. Anda dapat mengubah kebijakan kunci kapan saja dengan menggunakan PutKeyPolicyoperasi.

Saat Anda membuat kunci KMS diAWS Management Console, Anda juga membuat kebijakan utamanya. Saat Anda memilih identitas di bagian Administrator Kunci dan Pengguna Kunci, AWS KMS tambahkan pernyataan kebijakan untuk identitas tersebut ke kebijakan kunci kunci KMS.

Bagian Pengguna Kunci juga memungkinkan Anda menambahkan akun eksternal sebagai pengguna kunci.



Saat Anda memasukkan ID akun dari akun eksternal, AWS KMS akan menambahkan dua pernyataan untuk kebijakan kunci. Tindakan ini hanya mempengaruhi kebijakan kunci. Pengguna dan peran di akun eksternal tidak dapat menggunakan kunci KMS sampai Anda melampirkan kebijakan IAM untuk memberi mereka beberapa atau semua izin ini.

Pernyataan kebijakan kunci pertama memberikan izin akun eksternal untuk menggunakan kunci KMS dalam operasi kriptografi.

> **Note**
>
> Contoh berikut mewakili kebijakan kunci sampel untuk berbagi kunci KMS dengan akun lain. Ganti contoh`Sid`,`Principal`, dan `Action` nilai dengan nilai yang valid untuk tujuan penggunaan kunci KMS Anda.

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

Pernyataan kebijakan kunci kedua memungkinkan akun eksternal untuk membuat, melihat, dan mencabut hibah pada kunci KMS, tetapi hanya jika permintaan berasal dari AWSlayanan yang terintegrasi dengannya. AWS KMS Izin ini memungkinkan AWS layanan lain yang mengenkripsi data pengguna untuk menggunakan kunci KMS.

Izin ini dirancang untuk kunci KMS yang mengenkripsi data pengguna dalam AWS layanan, seperti Amazon. WorkMail Layanan ini biasanya menggunakan hibah untuk mendapatkan izin yang mereka butuhkan untuk menggunakan kunci KMS atas nama pengguna. Untuk detailnya, lihat Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS.

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

Jika izin ini tidak memenuhi kebutuhan Anda, Anda dapat mengeditnya di tampilan kebijakan konsol atau dengan menggunakan PutKeyPolicyoperasi. Anda dapat menentukan pengguna dan peran eksternal tertentu, alih-alih memberikan izin ke akun eksternal. Anda dapat mengubah tindakan yang ditentukan oleh kebijakan. Anda juga dapat menggunakan syarat kebijakan global dan AWS KMS untuk menyempurnakan izin.

Mengizinkan penggunaan kunci KMS eksternal dengan Layanan AWS

Anda dapat memberi pengguna izin akun yang berbeda untuk menggunakan kunci KMS Anda dengan layanan yang terintegrasi dengannyaAWS KMS. Misalnya, pengguna di akun eksternal dapat menggunakan kunci KMS Anda untuk mengenkripsi objek di bucket Amazon S3 atau untuk mengenkripsi rahasia yang mereka simpan. AWS Secrets Manager

Kebijakan kunci harus memberikan izin kepada pengguna eksternal atau akun pengguna eksternal untuk menggunakan kunci KMS. Selain itu, Anda perlu melampirkan kebijakan IAM ke identitas yang memberikan izin pengguna untuk menggunakan. Layanan AWS Layanan ini mungkin juga mengharuskan pengguna memiliki izin tambahan dalam kebijakan kunci atau kebijakan IAM. Untuk daftar izin yang Layanan AWS diperlukan pada kunci terkelola pelanggan, lihat topik Perlindungan Data di bagian Keamanan panduan pengguna atau panduan pengembang untuk layanan.

Menggunakan kunci KMS di akun lain

Jika Anda memiliki izin untuk menggunakan kunci KMS yang berbedaAkun AWS, Anda dapat menggunakan kunci KMS diAWS Management Console, AWS SDK, AWS CLI dan. AWS Tools for PowerShell

Untuk mengidentifikasi kunci KMS di akun yang berbeda dalam perintah shell atau permintaan API, gunakan pengidentifikasi kunci berikut.

- Untuk operasi kriptografi, DescribeKey, dan GetPublicKey, gunakan kunci ARN atau alias ARN dari kunci KMS.
- Untuk CreateGrant, GetKeyRotationStatus, ListGrants, dan RevokeGrant, gunakan tombol ARN dari tombol KMS.

Jika Anda hanya memasukkan ID kunci atau nama alias, AWS asumsikan kunci KMS ada di akun Anda.

AWS KMSKonsol tidak menampilkan kunci KMS di akun lain, bahkan jika Anda memiliki izin untuk menggunakannya. Juga, daftar kunci KMS yang ditampilkan di konsol AWS layanan lain tidak termasuk kunci KMS di akun lain.

Untuk menentukan kunci KMS di akun yang berbeda di konsol AWS layanan, Anda harus memasukkan kunci ARN atau alias ARN dari kunci KMS. Pengidentifikasi kunci yang diperlukan bervariasi menurut layanan, dan mungkin berbeda antara konsol layanan dan operasi API-nya. Untuk detailnya, lihat dokumentasi layanan.

Menggunakan peran tertaut layanan untuk AWS KMS

AWS Key Management Service menggunakan peran tertaut layanan AWS Identity and Access Management(IAM)https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_terms-and-concepts.html#iam-term-service-linked-role. Peran tertaut layanan adalah tipe IAM role unik yang

tertaut langsung ke AWS KMS. Peran yang terhubung dengan layanan ditentukan oleh AWS KMS dan mencakup semua izin yang diperlukan layanan untuk menghubungi layanan AWS lainnya atas nama Anda.

Peran tertaut layanan memudahkan penyiapan AWS KMS karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS KMS Peran ini menentukan izin peran tertaut layanannya, dan kecuali ditentukan lain, hanya AWS KMS dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran terhubung layanan hanya setelah pertama kali menghapus sumber daya terkait. Ini melindungi sumber daya AWS KMS karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran yang terhubung dengan layanan, lihat Layanan AWS yang Berfungsi dengan IAM dan cari layanan yang memiliki Ya di kolom Peran yang Terhubung dengan Layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran yang terhubung dengan layanan untuk penyimpanan kunci kustom AWS KMS

AWS KMSmenggunakan peran terkait layanan bernama AWSServiceRoleForKeyManagementServiceCustomKeyStoresuntuk mendukung penyimpanan kunci kustom. Peran yang terhubung dengan layanan ini memberikan izin AWS KMS untuk melihat cluster AWS CloudHSM Anda dan membuat infrastruktur jaringan untuk mendukung koneksi antara penyimpanan kunci kustom Anda dan kluster AWS CloudHSM-nya. AWS KMS membuat peran ini hanya ketika Anda membuat penyimpanan kunci kustom. Anda tidak dapat membuat peran yang terhubung dengan layanan ini secara langsung.

AWSServiceRoleForKeyManagementServiceCustomKeyStoresPeran terkait layanan percaya `cks.kms.amazonaws.com` untuk mengambil peran tersebut. Akibatnya, hanya AWS KMS yang dapat mengasumsikan peran yang terhubung dengan layanan ini.

Izin dalam peran terbatas pada tindakan yang dilakukan AWS KMS untuk menghubungkan penyimpanan kunci kustom kluster AWS CloudHSM. Ini tidak memberi AWS KMS izin tambahan apa pun. Misalnya, AWS KMS tidak memiliki izin untuk membuat, mengelola, atau menghapus perangkat klaster, HSM, atau pencadangan AWS CloudHSM.

Untuk informasi selengkapnya tentang AWSServiceRoleForKeyManagementServiceCustomKeyStoresperan, termasuk daftar izin dan instruksi tentang cara melihat peran, mengedit deskripsi peran, menghapus peran, dan AWS KMS membuatnya ulang untuk Anda, lihat. Mengizinkan AWS KMS mengelola sumber daya AWS CloudHSM Amazon EC2

Izin peran yang terhubung dengan layanan untuk kunci multi-Wilayah AWS KMS

AWS KMSmenggunakan peran terkait layanan bernama AWSServiceRoleForKeyManagementServiceMultiRegionKeysuntuk mendukung kunci Multi-wilayah. Peran yang terhubung dengan layanan ini memberikan izin AWS KMS untuk menyinkronkan perubahan apa pun ke materi kunci dari kunci utama multi-Wilayah untuk kunci replika. AWS KMS membuat peran ini hanya ketika Anda membuat Kunci utama multi-Wilayah. Anda tidak dapat membuat peran yang terhubung dengan layanan ini secara langsung.

AWSServiceRoleForKeyManagementServiceMultiRegionKeysPeran terkait layanan percaya `mrk.kms.amazonaws.com` untuk mengambil peran tersebut. Akibatnya, hanya AWS KMS yang dapat mengasumsikan peran yang terhubung dengan layanan ini. Izin dalam peran terbatas pada tindakan yang dilakukan AWS KMS untuk menjaga materi kunci dalam kunci multi-Wilayah terkait disinkronkan. Ini tidak memberi AWS KMS izin tambahan apa pun.

Untuk informasi selengkapnya tentang AWSServiceRoleForKeyManagementServiceMultiRegionKeysperan, termasuk daftar izin dan instruksi tentang cara melihat peran, mengedit deskripsi peran, menghapus peran, dan AWS KMS membuatnya ulang untuk Anda, lihat. Mengotorisasi AWS KMS untuk menyinkronkan kunci multi-Wilayah

AWS KMS memperbarui pada kebijakan terkelola AWS

Lihat detail tentang pembaruan pada kebijakan terkelola AWS untuk AWS KMS karena layanan ini mulai melacak perubahan tersebut. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan RSS feed pada halaman AWS KMS Riwayat dokumen.

Perubahan	Deskripsi	Tanggal
AWSKeyManagementServiceCustomKeyStoresService	AWS KMSmenambahkan`ec2:Descr`	10 November 2023

Perubahan	Deskripsi	Tanggal
RolePolicy – Pembaruan pada kebijakan yang sudah ada	`ibeVpcs ,ec2:Descr ibeNetworkAcls` , dan `ec2:DescribeNetwor kInterfaces` izin untuk memantau perubahan di VPC yang berisi cluster AWS CloudHSM Anda sehingga dapat memberikan pesan kesalahan AWS KMS yang jelas jika terjadi kegagalan.	
AWS KMS mulai melacak perubahan	AWS KMS mulai melacak perubahan untuk kebijakan terkelola AWS	10 November 2023

Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS

AWS Key Management Service (AWS KMS) mendukung pertukaran kunci pasca-kuantum hibrida untuk protokol enkripsi jaringan Keamanan Lapisan Pengangkutan (TLS). Anda dapat menggunakan opsi TLS ini ketika Anda terhubung ke titik akhir API AWS KMS. Kami menawarkan fitur ini sebelum algoritme pasca-kuantum distandardisasi, sehingga Anda dapat mulai menguji efek protokol pertukaran kunci ini pada panggilan AWS KMS. Fitur pertukaran kunci pasca-kuantum hibrida opsional ini setidaknya seaman enkripsi TLS yang kami gunakan saat ini dan cenderung memberikan manfaat keamanan jangka panjang tambahan. Namun, fitur-fitur tersebut memengaruhi latensi dan throughput dibandingkan dengan protokol pertukaran kunci klasik yang digunakan saat ini.

Data yang Anda kirimkan ke AWS Key Management Service (AWS KMS) dilindungi dalam transit oleh enkripsi yang disediakan oleh koneksi Keamanan Lapisan Pengangkutan (TLS). Cipher suite klasik yang didukung AWS KMS untuk sesi TLS membuat serangan brute force pada mekanisme pertukaran kunci yang tidak layak dengan teknologi saat ini. Namun, jika komputasi kuantum skala besar menjadi praktis di masa depan, cipher suite klasik yang digunakan dalam mekanisme pertukaran kunci TLS akan rentan terhadap serangan ini. Jika Anda terngah mengembangkan aplikasi yang mengandalkan kerahasiaan data jangka panjang melalui sambungan TLS, Anda harus mempertimbangkan rencana untuk bermigrasi ke kriptografi pasca-kuantum sebelum komputer

kuantum skala besar tersedia untuk digunakan. AWS berupaya mempersiapkan masa depan ini, dan kami juga ingin agar Anda mempersiapkannya dengan matang.

Guna melindungi data yang dienkripsi saat ini terhadap potensi serangan masa depan, AWS berpartisipasi dengan komunitas kriptografi dalam pengembangan algoritme tahan-kuantum atau pasca-kuantum. Kami telah menerapkan suite cipher pertukaran kunci pasca-kuantum hibrida AWS KMS yang menggabungkan elemen klasik dan pasca-kuantum untuk memastikan bahwa koneksi TLS Anda setidaknya sekuat dengan suite cipher klasik.

Cipher suite hibrida ini tersedia untuk digunakan pada beban kerja produksi Anda di sebagian besar Wilayah AWS. Namun, karena karakteristik performa dan persyaratan bandwidth cipher suite hibrida berbeda dari mekanisme pertukaran kunci klasik, kami menyarankan Anda mengujinya di panggilan AWS KMS API dalam kondisi yang berbeda.

Umpan Balik

Seperti biasa, kami menyambut baik umpan balik dan partisipasi Anda dalam repositori sumber terbuka kami. Kami terutama ingin mendengar bagaimana infrastruktur Anda berinteraksi dengan varian lalu lintas TLS yang baru ini.

- Untuk memberikan umpan balik tentang topik ini, gunakan tautan Umpan Balik di sudut kanan atas halaman ini.
- Kami sedang mengembangkan suite cipher hybrid ini di open source di s2n-tlsrepositori di. GitHub Untuk memberikan umpan balik tentang kegunaan rangkaian sandi, atau berbagi kondisi atau hasil pengujian baru, buat masalah di repositori. s2n-tls
- Kami sedang menulis contoh kode untuk menggunakan TLS pasca-kuantum hibrida dengan AWS KMS di repositori. aws-kms-pq-tls-example GitHub Untuk mengajukan pertanyaan atau berbagi ide tentang mengonfigurasi klien HTTP Anda atau klien AWS KMS untuk menggunakan cipher suite hibrida, buat masalah di repositori aws-kms-pq-tls-example.

Mendukung Wilayah AWS

TLS pasca-kuantum untuk AWS KMS tersedia di semua Wilayah AWS yang AWS KMS mendukung kecuali untuk China (Beijing) dan China (Ningxia).

> **Note**
>
> AWS KMStidak mendukung TLS pasca-kuantum hibrida untuk titik akhir FIPS di. AWS GovCloud (US)

Untuk daftar AWS KMS titik akhir untuk masing-masingWilayah AWS, lihat AWS Key Management Servicetitik akhir dan kuota di. Referensi Umum Amazon Web Services Untuk informasi tentang titik akhir FIPS, lihat titik akhir FIPS di. Referensi Umum Amazon Web Services

Tentang pertukaran kunci pasca-kuantum hibrida di TLS

AWS KMS mendukung cipher suite pertukaran kunci pasca-kuantum hibrida. Anda dapat menggunakan AWS SDK for Java 2.x dan AWS Common Runtime pada sistem Linux untuk mengkonfigurasi klien HTTP yang menggunakan cipher suite ini. Kemudian, setiap kali Anda terhubung ke AWS KMS titik akhir dengan klien HTTP Anda, suite cipher hybrid digunakan.

Klien HTTP ini menggunakan s2n-tls, yang merupakan implementasi open source dari protokol TLS. Suite cipher hybrid yang s2n-tls menggunakan diimplementasikan hanya untuk pertukaran kunci, bukan untuk enkripsi data langsung. Selama pertukaran kunci, klien dan server menghitung kunci yang akan mereka gunakan untuk mengenkripsi dan mendekripsi data pada kabel.

Algoritma yang s2n-tls digunakan adalah hibrida yang menggabungkan Elliptic Curve Diffie-Hellman (ECDH), algoritma pertukaran kunci klasik yang digunakan saat ini di TLS, dengan Kyber, enkripsi kunci publik dan algoritma pembentukan kunci yang telah ditetapkan oleh Institut Nasional untuk Standar dan Teknologi (NIST) sebagai algoritma perjanjian kunci pasca-kuantum standar pertama. Hibrida ini menggunakan masing-masing algoritma secara independen untuk menghasilkan kunci. Selanjutnya menggabungkan dua kunci kriptografi. Dengans2n-tls, Anda dapat mengonfigurasi klien HTTP untuk memilih TLS pasca-kuantum, yang menempatkan ECDH dengan yang Kyber pertama dalam daftar preferensi. Algoritme pertukaran kunci klasik disertakan dalam daftar preferensi untuk memastikan kompatibilitasnya, namun lebih rendah dalam urutan preferensi.

Jika penelitian yang sedang berlangsung mengungkapkan bahwa Kyber algoritme tidak memiliki kekuatan pasca-kuantum yang diantisipasi, kunci hibrida setidaknya masih sekuat kunci ECDH tunggal yang saat ini digunakan. Sampai penelitian tentang algoritma pasca-kuantum selesai, kami merekomendasikan menggunakan algoritma hibrida, daripada menggunakan algoritma pasca-kuantum saja.

Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS

Anda dapat menggunakan TLS pasca-kuantum hibrida untuk panggilan Anda ke AWS KMS. Saat menyiapkan lingkungan pengujian klien HTTP Anda, perhatikan informasi berikut:

Enkripsi dalam Transit

Suite cipher hybrid di hanya s2n-tls digunakan untuk enkripsi dalam perjalanan. Mereka melindungi data Anda saat bepergian dari klien Anda ke AWS KMS titik akhir. AWS KMStidak menggunakan cipher suite ini untuk mengenkripsi data di bawah. AWS KMS keys

Sebaliknya, ketika AWS KMS mengenkripsi data Anda di bawah kunci KMS, ia menggunakan kriptografi simetris dengan kunci 256-bit dan Standar Enkripsi Lanjutan dalam algoritma Galois Counter Mode (AES-GCM), yang sudah tahan kuantum. Masa depan teoritis, serangan komputasi kuantum skala besar pada ciphertext yang dibuat di bawah kunci AES-GCM 256-bit mengurangi keamanan yang efektif dari kunci tersebut ke 128 bit. Tingkat keamanan ini cukup untuk membuat serangan brute force pada ciphertext AWS KMS yang tidak layak.

Sistem yang Didukung

Penggunaan suite cipher hybrid di saat s2n-tls ini hanya didukung pada sistem Linux. Selain itu, cipher suite ini didukung hanya dalam SDK yang mendukung Waktu Aktif Umum AWS, seperti AWS SDK for Java 2.x. Sebagai contoh, lihat Cara mengonfigurasi TLS pasca-kuantum hibrida.

AWS KMSTitik akhir

Saat menggunakan cipher suite hibrida, gunakan titik akhir AWS KMS standar. Suite sandi hibrida di tidak kompatibel dengan titik akhir s2n-tls yang divalidasi FIPS 140-2 untuk. AWS KMS

Saat Anda mengonfigurasi klien HTTP untuk memilih koneksi TLS pasca-kuantums2n-tls, sandi pasca-kuantum adalah yang pertama dalam daftar preferensi sandi. Namun, daftar preferensi ini mencakup cipher klasik dan non-hibrida yang lebih rendah dalam urutan preferensi untuk kompatibilitas. Saat Anda mengonfigurasi klien HTTP untuk memilih TLS pasca-kuantum dengan titik akhir yang divalidasi AWS KMS FIPS 140-2, s2n-tls menegosiasikan cipher pertukaran kunci non-hybrid klasik.

Untuk daftar AWS KMS titik akhir untuk masing-masingWilayah AWS, lihat AWS Key Management Servicetitik akhir dan kuota di. Referensi Umum Amazon Web Services Untuk informasi tentang titik akhir FIPS, lihat titik akhir FIPS di. Referensi Umum Amazon Web Services

Kinerja yang Diharapkan

Pengujian benchmark awal kami menunjukkan bahwa suite cipher hybrid lebih lambat daripada suite cipher TLS klasik. s2n-tls Efeknya bervariasi, tergantung profil jaringan, kecepatan CPU, jumlah core, dan tingkat panggilan Anda. Untuk hasil tes kinerja, lihat Cara menyetel TLS untuk kriptografi pasca-kuantum hibrida dengan Kyber.

Cara mengonfigurasi TLS pasca-kuantum hibrida

Dalam prosedur ini, tambahkan dependensi Maven untuk AWS Common Runtime HTTP Client. Selanjutnya, konfigurasikan klien HTTP yang lebih memilih TLS pasca-kuantum. Kemudian, buat klien AWS KMS yang menggunakan klien HTTP.

Untuk melihat contoh kerja lengkap tentang mengonfigurasi dan menggunakan TLS pasca-kuantum hibrida denganAWS KMS, lihat repositori. aws-kms-pq-tls-example

> **Note**
>
> Klien HTTP Runtime AWS Umum, yang telah tersedia sebagai pratinjau, tersedia secara umum pada Februari 2023. Dalam rilis itu, `tlsCipherPreference` kelas dan parameter `tlsCipherPreference()` metode digantikan oleh parameter `postQuantumTlsEnabled()` metode. Jika Anda menggunakan contoh ini selama pratinjau, Anda perlu memperbarui kode Anda.

1. Tambahkan klien Common Runtime AWS untuk dependensi Maven Anda. Sebaiknya gunakan versi terbaru yang tersedia.

   Misalnya, pernyataan ini menambahkan versi `2.20.0` klien AWS Common Runtime ke dependensi Maven Anda.

   ```
   <dependency>
       <groupId>software.amazon.awssdk</groupId>
       <artifactId>aws-crt-client</artifactId>
       <version>2.20.0</version>
   </dependency>
   ```

2. Untuk mengaktifkan suite cipher pasca-kuantum hibrida, tambahkan AWS SDK for Java 2.x untuk proyek Anda dan menginisialisasinya. Kemudian aktifkan suite sandi pasca-kuantum hibrida pada klien HTTP Anda seperti yang ditunjukkan pada contoh berikut.

Kode ini menggunakan parameter `postQuantumTlsEnabled()` metode untuk mengonfigurasi klien HTTP runtime AWS umum yang lebih menyukai rangkaian sandi pasca-kuantum hibrida yang direkomendasikan, ECDH dengan Kyber. Kemudian menggunakan klien HTTP yang dikonfigurasi untuk membangun instance klien AWS KMS asinkron,. `KmsAsyncClient` Setelah kode ini selesai, semua permintaan AWS KMSAPI pada `KmsAsyncClient` instance menggunakan TLS pasca-kuantum hibrida.

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

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

3. Uji AWS KMS panggilan Anda dengan TLS pasca-kuantum hibrida.

   Ketika Anda memanggil operasi AWS KMS API pada klien AWS KMS yang telah dikonfigurasi, panggilan Anda ditransmisikan ke titik akhir AWS KMS menggunakan TLS pasca-kuantum hibrida. Untuk menguji konfigurasi Anda, panggil AWS KMS API, seperti`ListKeys`.

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

Menguji TLS pasca-kuantum hibrida dengan AWS KMS

Pertimbangkan menjalankan uji berikut dengan cipher suite hibrida pada aplikasi Anda yang memanggil AWS KMS.

- Jalankan uji beban dan tolok ukur. Cipher suite hibrida melakukan secara berbeda dari algoritme pertukaran kunci tradisional. Anda mungkin perlu menyesuaikan batas waktu koneksi untuk memungkinkan waktu handshake yang lebih lama. Jika Anda menjalankan di dalam sebuah fungsi AWS Lambda, perpanjang pengaturan waktu habis eksekusi.
- Coba hubungkan dari lokasi yang berbeda. Tergantung jalur jaringan yang dibutuhkan permintaannya, Anda mungkin menemukan bahwa host perantara, proksi, atau firewall dengan deep packet inspection (DPI) memblokir permintaan. Ini mungkin hasil dari penggunaan cipher suite baru di ClientHellobagian jabat tangan TLS, atau dari pesan pertukaran kunci yang lebih

besar. Jika Anda terkendala saat menyelesaikan masalah ini, tanyakan kepada tim keamanan atau administrator IT Anda untuk memperbarui konfigurasi yang relevan dan membuka blokir rangkaian penyandian TLS baru.

Pelajari lebih lanjut tentang TLS pasca-kuantum di AWS KMS

Untuk informasi lebih lanjut tentang penggunaan TLS pasca-kuantum hibrida di AWS KMS, lihat sumber daya berikut.

- Untuk mempelajari tentang kriptografi pasca-kuantum diAWS, termasuk tautan ke posting blog dan makalah penelitian, lihat Kriptografi Pasca-Kuantum.
- Untuk selengkapnyas2n-tls, lihat Memperkenalkans2n-tls, Implementasi dan Penggunaan s2n-tls TLS Open Source Baru.
- Untuk informasi tentang Klien HTTP Runtime AWS Umum, lihat Mengonfigurasi klien HTTP AWS berbasis CRT di Panduan Pengembang. AWS SDK for Java 2.x
- Untuk informasi tentang proyek kriptografi pasca-kuantum di National Institute for Standards and Technology (NIST), lihat Kriptografi Pasca Kuantum.
- Untuk informasi tentang standardisasi kriptografi pasca-kuantum NIST, lihat Standardisasi Kriptografi Pasca-Kuantum.

Menentukan akses ke AWS KMS keys

Untuk menentukan sepenuhnya siapa atau apa yang saat ini memiliki akses ke suatuAWS KMS key, Anda harus memeriksa kebijakan kunci KMS, semua hibah yang berlaku untuk kunci KMS, dan potensi semua AWS Identity and Access Management (IAM) kebijakan. Anda dapat melakukan ini untuk menentukan ruang lingkup potensi penggunaan kunci KMS, atau untuk membantu Anda memenuhi persyaratan kepatuhan atau audit. Topik berikut dapat membantu Anda menghasilkan daftar lengkap AWS kepala sekolah (identitas) yang saat ini memiliki akses ke kunci KMS.

Topik

- Memeriksa kebijakan kunci
- Memeriksa kebijakan IAM
- Memeriksa pemberian izin
- Memecahkan masalah akses kunci

Memeriksa kebijakan kunci

Kebijakan utama adalah cara utama untuk mengontrol akses ke kunci KMS. Setiap kunci KMS memiliki persis satu kebijakan utama.

Jika kebijakan kunci terdiri dari atau menyertakan kebijakan kunci default, kebijakan kunci memungkinkan administrator IAM di akun untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS. Selain itu, jika kebijakan kunci memberikan Akun AWS izin lain untuk menggunakan kunci KMS, administrator IAM di akun eksternal dapat menggunakan kebijakan IAM untuk mendelegasikan izin tersebut. Untuk menentukan daftar lengkap kepala sekolah yang dapat mengakses kunci KMS, periksa kebijakan IAM.

Untuk melihat kebijakan kunci kunci yang dikelola AWS KMS pelanggan atau Kunci yang dikelola AWSdi akun Anda, gunakan AWS Management Console atau GetKeyPolicyoperasi di AWS KMS API. Untuk melihat kebijakan kunci, Anda harus memiliki `kms:GetKeyPolicy` izin untuk kunci KMS. Untuk petunjuk untuk melihat kebijakan kunci untuk kunci KMS, lihatthe section called “Melihat kebijakan kunci”.

Periksa dokumen kebijakan kunci dan perhatikan semua perwakilan yang ditentukan dalam setiap elemen `Principal` pernyataan kebijakan. Dalam pernyataan kebijakan dengan `Allow` efek, pengguna IAM, peran IAM, dan Akun AWS `Principal` elemen memiliki akses ke kunci KMS ini.

> **Note**
>
> Jangan menyetel Principal ke tanda bintang (*) dalam pernyataan kebijakan kunci apa pun yang mengizinkan izin kecuali Anda menggunakan kondisi untuk membatasi kebijakan utama. Tanda bintang memberikan setiap identitas di setiap Akun AWS izin untuk menggunakan kunci KMS, kecuali pernyataan kebijakan lain secara eksplisit menyangkalnya. Pengguna lain Akun AWS dapat menggunakan kunci KMS Anda setiap kali mereka memiliki izin yang sesuai di akun mereka sendiri.

Contoh berikut menggunakan pernyataan kebijakan yang ditemukan di kebijakan kunci default untuk menunjukkan cara melakukannya.

Example Pernyataan kebijakan 1

```
{
  "Sid": "Enable IAM User Permissions",
```

```
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
  "Resource": "*"
}
```

Dalam pernyataan kebijakan 1, `arn:aws:iam::111122223333:root` adalah pokok AWS akun yang mengacu pada Akun AWS 111122223333. (Ini bukan pengguna root akun.) Secara default, pernyataan kebijakan seperti ini disertakan dalam dokumen kebijakan kunci saat Anda membuat kunci KMS baru denganAWS Management Console, atau membuat kunci KMS baru secara terprogram tetapi tidak menyediakan kebijakan kunci.

Dokumen kebijakan kunci dengan pernyataan yang memungkinkan akses ke Akun AWS mengaktifkan kebijakan IAM di akun untuk memungkinkan akses ke kunci KMS. Ini berarti bahwa pengguna dan peran dalam akun mungkin memiliki akses ke kunci KMS bahkan jika mereka tidak secara eksplisit terdaftar sebagai prinsipal dalam dokumen kebijakan utama. Berhati-hatilah untuk memeriksa semua kebijakan IAM di semua yang Akun AWS terdaftar sebagai prinsipal untuk menentukan apakah mereka mengizinkan akses ke kunci KMS ini.

Example Pernyataan kebijakan 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": "*"
}
```

Dalam pernyataan kebijakan 2, `arn:aws:iam::111122223333:role/KMSKeyAdmins` mengacu pada peran IAM bernama KMS KeyAdmins di Akun AWS 111122223333. Pengguna yang berwenang untuk mengambil peran ini diizinkan untuk melakukan tindakan yang tercantum dalam pernyataan kebijakan, yang merupakan tindakan administratif untuk mengelola kunci KMS.

Example Kebijakan pernyataan 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": "*"
}
```

Dalam pernyataan kebijakan 3, `arn:aws:iam::111122223333:role/EncryptionApp` mengacu pada peran IAM yang disebutkan EncryptionApp dalam Akun AWS 111122223333. Kepala sekolah yang berwenang untuk mengambil peran ini diizinkan untuk melakukan tindakan yang tercantum dalam pernyataan kebijakan, yang mencakup operasi kriptografi untuk kunci KMS enkripsi simetris.

Example Kebijakan pernyataan 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}}
}
```

Dalam pernyataan kebijakan 4, `arn:aws:iam::111122223333:role/EncryptionApp` mengacu pada peran IAM yang disebutkan EncryptionApp dalam Akun AWS 111122223333. Kepala sekolah yang berwenang mengambil peran ini diizinkan untuk melakukan tindakan yang tercantum dalam pernyataan kebijakan. Tindakan ini, bila dikombinasikan dengan tindakan yang diizinkan dalam pernyataan kebijakan Contoh 3, adalah tindakan yang diperlukan untuk mendelegasikan penggunaan kunci KMS ke sebagian besar AWSlayanan yang terintegrasi dengan AWS KMS, khususnya layanan yang menggunakan hibah. GrantIsForAWSResourceNilai kms: dalam `Condition` elemen memastikan bahwa delegasi hanya diperbolehkan ketika delegasi adalah AWS layanan yang terintegrasi dengan AWS KMS dan menggunakan hibah untuk otorisasi.

Untuk mempelajari semua cara yang berbeda, Anda dapat menentukan perwakilan dalam dokumen kebijakan kunci, lihat Menentukan Perwakilan di Panduan Pengguna IAM.

Untuk mempelajari selengkapnya tentang kebijakan kunci AWS KMS, lihat Kebijakan utama di AWS KMS.

Memeriksa kebijakan IAM

Selain kebijakan dan hibah utama, Anda juga dapat menggunakan kebijakan IAM untuk mengizinkan akses ke kunci KMS. Untuk informasi selengkapnya tentang bagaimana kebijakan IAM dan kebijakan kunci bekerja sama, lihat Memecahkan masalah akses kunci.

Untuk menentukan prinsipal mana yang saat ini memiliki akses ke kunci KMS melalui kebijakan IAM, Anda dapat menggunakan alat IAM Policy Simulator berbasis browser, atau Anda dapat membuat permintaan ke API IAM.

Cara memeriksa kebijakan IAM

- Menguji kebijakan IAM dengan simulator kebijakan IAM
- Menguji kebijakan IAM dengan API IAM

Menguji kebijakan IAM dengan simulator kebijakan IAM

Simulator Kebijakan IAM dapat membantu Anda mempelajari prinsip mana yang memiliki akses ke kunci KMS melalui kebijakan IAM.

Untuk menggunakan simulator kebijakan IAM untuk menentukan akses ke kunci KMS

1. Masuk ke AWS Management Console lalu buka Simulator Kebijakan IAM di https://policysim.aws.amazon.com/.

2. Di panel Pengguna, Grup, dan Peran, pilih pengguna, grup, atau peran yang kebijakannya ingin disimulasikan.
3. (Opsional) Hapus kotak centang di samping kebijakan yang ingin dihilangkan dari simulasi. Untuk mensimulasikan semua kebijakan, biarkan semua kebijakan dipilih.
4. Di panel Simulator Kebijakan, lakukan hal berikut:
   a. Untuk Pilih layanan, pilih Layanan Manajemen Kunci.
   b. Untuk mensimulasikan tindakan AWS KMS tertentu, untuk Pilih tindakan, pilih tindakan yang akan disimulasikan. Untuk mensimulasikan semua tindakan AWS KMS, pilih Pilih Semua.
5. (Opsional) Simulator Kebijakan mensimulasikan akses ke semua kunci KMS secara default. Untuk mensimulasikan akses ke kunci KMS tertentu, pilih Pengaturan Simulasi dan kemudian ketik Nama Sumber Daya Amazon (ARN) dari kunci KMS untuk disimulasikan.
6. Pilih Jalankan Simulasi.

Anda bisa melihat hasil simulasi di bagian Hasil. Ulangi langkah 2 hingga 6 untuk setiap pengguna, grup, dan peran dalamAkun AWS.

Menguji kebijakan IAM dengan API IAM

Anda dapat menggunakan API IAM untuk memeriksa kebijakan IAM secara terprogram. Langkah-langkah berikut memberikan gambaran umum tentang cara melakukannya:

1. Untuk setiap yang Akun AWS terdaftar sebagai prinsipal dalam kebijakan kunci (yaitu, setiap prinsipal AWS akun yang ditentukan dalam format ini:`"Principal": {"AWS": "arn:aws:iam::111122223333:root"}`), gunakan ListUsersdan ListRolesoperasi di API IAM untuk mendapatkan semua pengguna dan peran dalam akun.
2. Untuk setiap pengguna dan peran dalam daftar, gunakan SimulatePrincipalPolicyoperasi di API IAM, meneruskan parameter berikut:
   - Untuk `PolicySourceArn`, tentukan Amazon Resource Name (ARN) pengguna atau peran dari daftar Anda. Anda hanya dapat menentukan satu `PolicySourceArn` untuk setiap `SimulatePrincipalPolicy` permintaan, jadi Anda harus memanggil operasi ini beberapa kali, sekali untuk setiap pengguna dan peran dalam daftar Anda.
   - Untuk daftar `ActionNames`, tentukan setiap tindakan API AWS KMS yang akan mensimulasikan. Untuk mensimulasikan semua tindakan API AWS KMS, gunakan `kms:*`. Untuk menguji tindakan API AWS KMS individu, awali setiap tindakan API dengan "`kms:`",

misalnya "`kms:ListKeys`". Untuk daftar lengkap tindakan AWS KMS API, lihat Tindakan di Referensi AWS Key Management Service API.

- (Opsional) Untuk menentukan apakah pengguna atau peran memiliki akses ke kunci KMS tertentu, gunakan `ResourceArns` parameter untuk menentukan daftar Nama Sumber Daya Amazon (ARN) kunci KMS. Untuk menentukan apakah pengguna atau peran memiliki akses ke kunci KMS apa pun, hilangkan parameter. `ResourceArns`

IAM merespons setiap permintaan `SimulatePrincipalPolicy` dengan keputusan evaluasi: `allowed`, `explicitDeny`, atau `implicitDeny`. Untuk setiap respons yang berisi keputusan evaluasi `allowed`, respons yang menyertakan nama operasi API AWS KMS tertentu yang diizinkan. Ini juga termasuk ARN kunci KMS yang digunakan dalam evaluasi, jika ada.

Memeriksa pemberian izin

Hibah adalah mekanisme lanjutan untuk menentukan izin yang AWS KMS dapat digunakan oleh Anda atau AWS layanan terintegrasi untuk menentukan bagaimana dan kapan kunci KMS dapat digunakan. Hibah dilampirkan ke kunci KMS, dan setiap hibah berisi kepala sekolah yang menerima izin untuk menggunakan kunci KMS dan daftar operasi yang diizinkan. Pemberian izin adalah alternatif untuk kebijakan kunci , dan berguna untuk kasus penggunaan tertentu. Untuk informasi selengkapnya, lihat Hibah di AWS KMS.

Untuk mendapatkan daftar hibah untuk kunci KMS, gunakan operasi. AWS KMS ListGrants Anda dapat memeriksa hibah untuk kunci KMS untuk menentukan siapa atau apa yang saat ini memiliki akses untuk menggunakan kunci KMS melalui hibah tersebut. Sebagai contoh, berikut ini representasi JSON bantuan yang diperoleh dari perintah list-grants dalam AWS CLI.

```
{"Grants": [{
  "Operations": ["Decrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "0d8aa621-43ef-4657-b29c-3752c41dc132",
  "RetiringPrincipal": "arn:aws:iam::123456789012:root",
  "GranteePrincipal": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-5d476fab",
  "GrantId": "dc716f53c93acacf291b1540de3e5a232b76256c83b2ecb22cdefa26576a2d3e",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151834E9,
  "Constraints": {"EncryptionContextSubset": {"aws:ebs:id": "vol-5cccfb4e"}}
}]}
```

Untuk mengetahui siapa atau apa yang memiliki akses untuk menggunakan tombol KMS, cari `"GranteePrincipal"` elemennya. Dalam contoh sebelumnya, perwakilan penerima adalah peran pengguna diasumsikan yang berhubungan dengan instans EC2 i-5d476fab. Infrastruktur EC2 menggunakan peran ini untuk melampirkan volume vol-5cccfb4e EBS terenkripsi ke instans. Dalam hal ini, peran infrastruktur EC2 memiliki izin untuk menggunakan kunci KMS karena sebelumnya Anda membuat volume EBS terenkripsi yang dilindungi oleh kunci KMS ini. Anda kemudian melampirkan volume ke instans EC2.

Berikut ini contoh lain dari representasi JSON pemberian izin yang diperoleh dari perintah list-grants dalam AWS CLI. Pada contoh berikut, perwakilan penerima adalah Akun AWS yang lain.

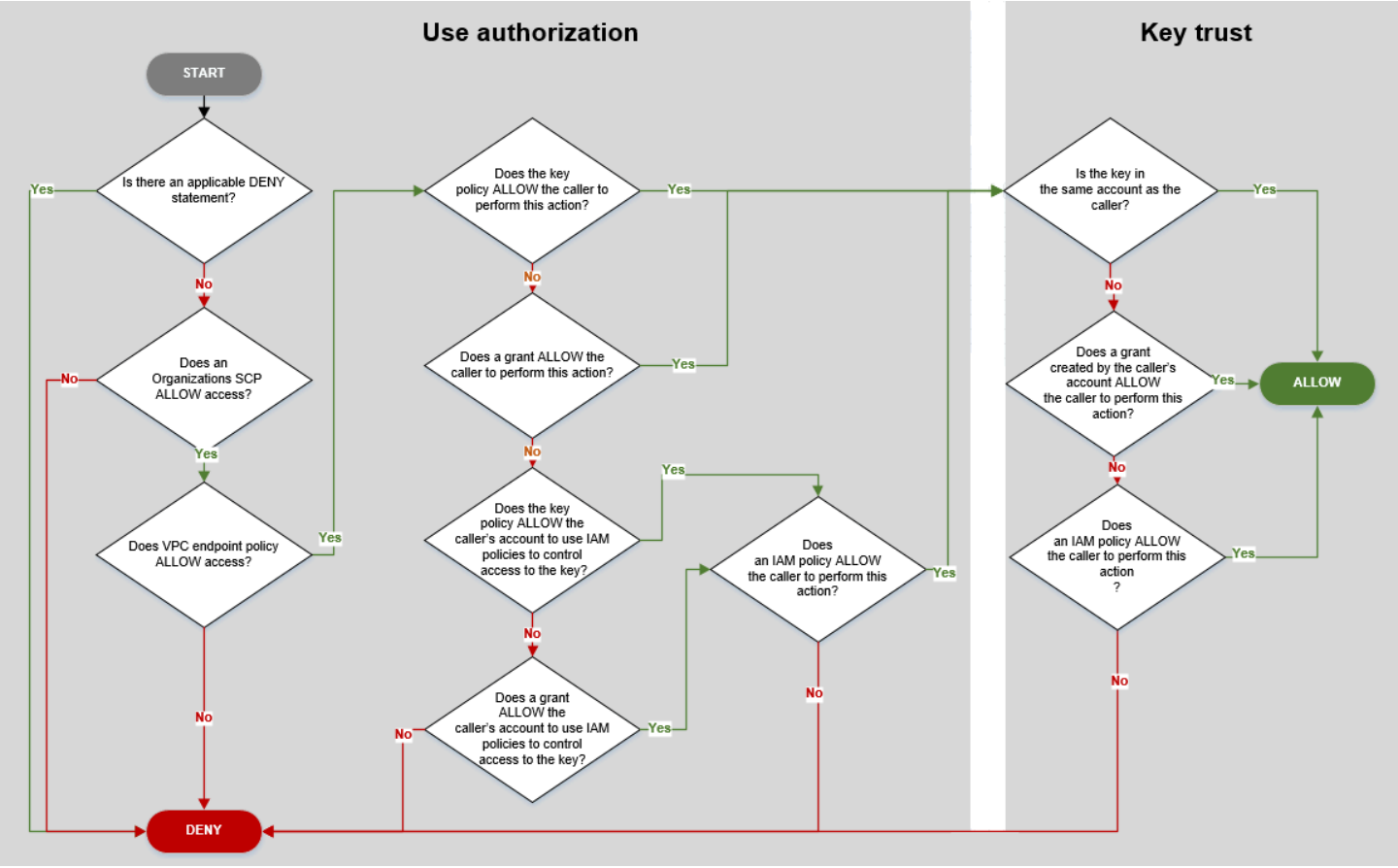
```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

Memecahkan masalah akses kunci

Saat mengotorisasi akses ke kunci KMS, AWS KMS evaluasi hal berikut:

- Kebijakan kunci yang dilampirkan pada kunci KMS. Kebijakan kunci selalu didefinisikan di Akun AWS dan Wilayah yang memiliki kunci KMS.
- Semua kebijakan IAM yang dilampirkan pada pengguna atau peran yang membuat permintaan. Kebijakan IAM yang mengatur penggunaan kunci KMS oleh prinsipal selalu didefinisikan dalam prinsipal. Akun AWS
- Semua hibah yang berlaku untuk kunci KMS.
- Jenis kebijakan lain yang mungkin berlaku untuk permintaan untuk menggunakan kunci KMS, seperti kebijakan kontrol AWS Organizations layanan dan kebijakan titik akhir VPC. Kebijakan ini bersifat opsional dan mengaktifkan semua tindakan secara default, namun Anda dapat menggunakannya untuk membatasi izin yang diberikan kepada perwakilan.

AWS KMSmengevaluasi mekanisme kebijakan ini bersama-sama untuk menentukan apakah akses ke kunci KMS diperbolehkan atau ditolak. Untuk melakukannya, AWS KMS menggunakan proses yang mirip dengan yang digambarkan dalam bagan alur berikut. Bagan alur berikut memberikan representasi visual dari proses evaluasi kebijakan.



Bagan alur ini dibagi menjadi dua bagian. Bagian-bagian ini tampaknya berurutan, tetapi biasanya dievaluasi pada waktu yang sama.

- Otorisasi penggunaan menentukan apakah Anda diizinkan untuk menggunakan kunci KMS berdasarkan kebijakan utama, kebijakan IAM, hibah, dan kebijakan lain yang berlaku.
- Kepercayaan kunci menentukan apakah Anda harus mempercayai kunci KMS yang diizinkan untuk Anda gunakan. Secara umum, Anda mempercayai sumber daya dalam Akun AWS. Namun, Anda juga dapat merasa yakin tentang menggunakan kunci KMS dalam hal yang berbeda Akun AWS jika hibah atau kebijakan IAM di akun Anda memungkinkan Anda untuk menggunakan kunci KMS.

Anda dapat menggunakan diagram alur ini untuk mengetahui mengapa penelepon diizinkan atau ditolak izin untuk menggunakan kunci KMS. Anda juga dapat menggunakannya untuk mengevaluasi kebijakan dan pemberian izin. Misalnya, bagan alur menunjukkan bahwa pemanggil dapat ditolak

aksesnya oleh pernyataan DENY eksplisit, atau dengan tidak adanya pernyataan ALLOW eksplisit, dalam kebijakan kunci, kebijakan IAM, atau pemberian izin.

Bagan alur dapat menjelaskan beberapa skenario izin umum.

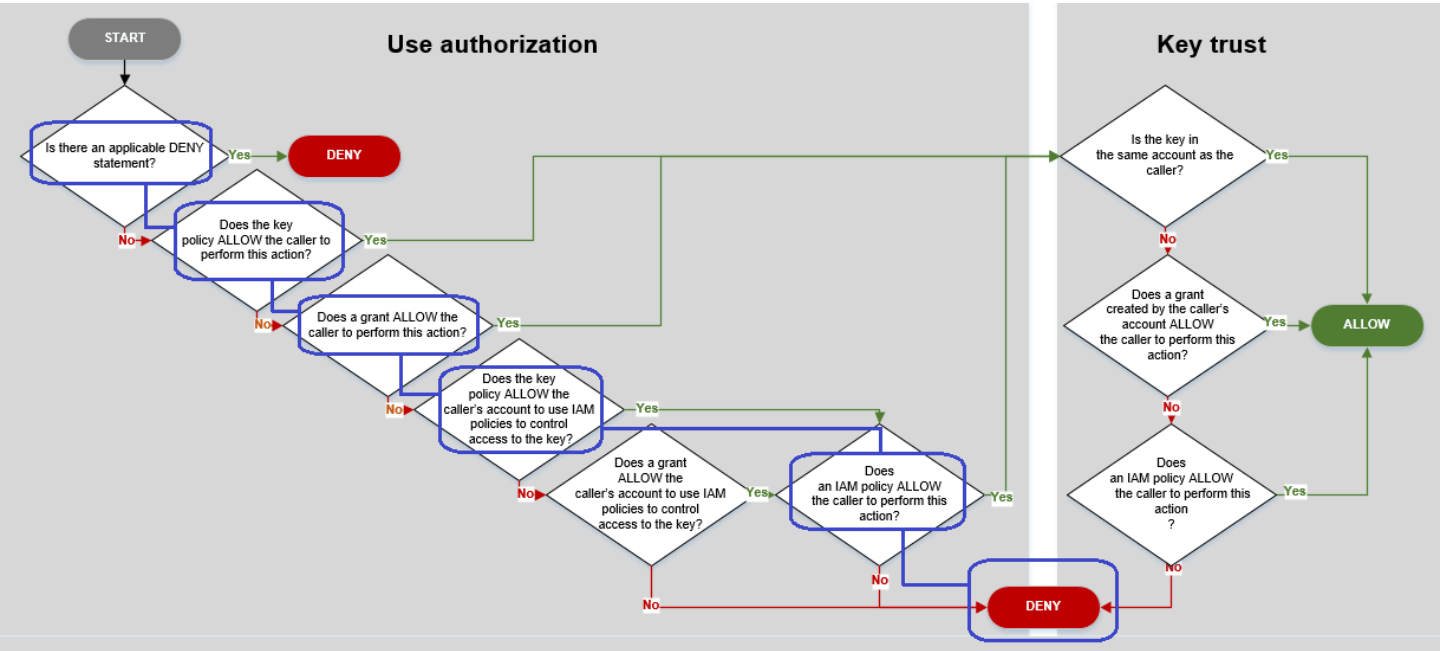
Contoh Izin

- Contoh 1: Pengguna ditolak akses ke kunci KMS di Akun AWS
- Contoh 2: Pengguna mengasumsikan peran dengan izin untuk menggunakan kunci KMS dalam yang berbeda Akun AWS

Contoh 1: Pengguna ditolak akses ke kunci KMS di Akun AWS

Alice adalah pengguna IAM di Akun AWS 111122223333. Dia ditolak akses ke kunci KMS dalam hal yang samaAkun AWS. Mengapa Alice tidak bisa menggunakan tombol KMS?

Dalam hal ini, Alice ditolak akses ke kunci KMS karena tidak ada kebijakan kunci, kebijakan IAM, atau hibah yang memberinya izin yang diperlukan. Kebijakan kunci kunci KMS memungkinkan Akun AWS untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS, tetapi tidak ada kebijakan IAM yang memberikan izin kepada Alice untuk menggunakan kunci KMS.



Pertimbangkan kebijakan yang relevan untuk contoh ini.

- Kunci KMS yang ingin digunakan Alice memiliki kebijakan kunci default. Kebijakan ini memungkinkan pemilik Akun AWS kunci KMS untuk menggunakan kebijakan IAM untuk

mengontrol akses ke kunci KMS. Kebijakan kunci ini memenuhi syarat Apakah kebijakan kunci MENGIZINKAN akun pemanggil menggunakan kebijakan IAM untuk mengontrol akses ke kunci? dalam bagan alur.

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

- Namun, tidak ada kebijakan kunci, kebijakan IAM, atau hibah yang memberikan izin kepada Alice untuk menggunakan kunci KMS. Oleh karena itu, Alice ditolak izinnya untuk menggunakan kunci KMS.

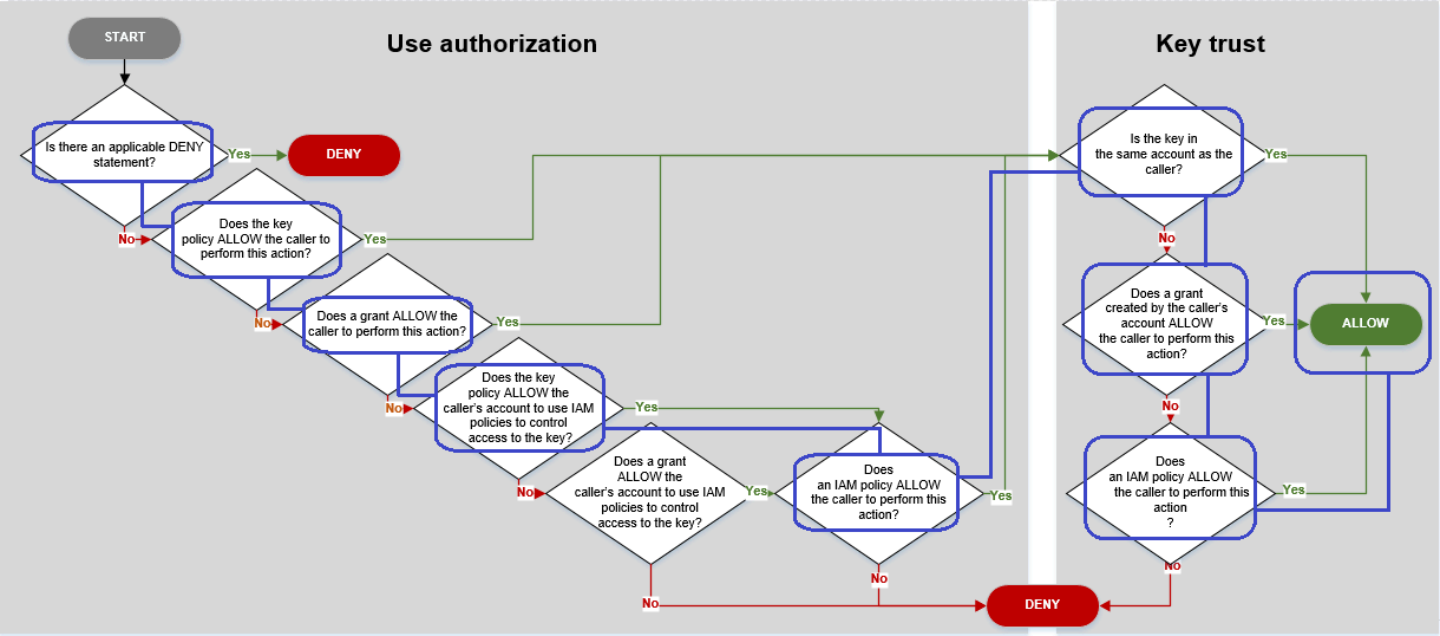
Contoh 2: Pengguna mengasumsikan peran dengan izin untuk menggunakan kunci KMS dalam yang berbeda Akun AWS

Bob adalah pengguna di akun 1 (111122223333). Dia diizinkan untuk menggunakan kunci KMS di akun 2 (444455556666) dalam operasi kriptografi. Bagaimana ini mungkin?

> **Tip**
>
> Saat mengevaluasi izin lintas akun, ingatlah bahwa kebijakan kunci ditentukan dalam akun kunci KMS. Kebijakan IAM ditentukan dalam akun pemanggil, bahkan ketika pemanggil berada dalam akun yang berbeda. Untuk detail tentang menyediakan akses lintas akun ke kunci KMS, lihat. Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS

- Kebijakan kunci untuk kunci KMS di akun 2 memungkinkan akun 2 menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS.

- Kebijakan kunci untuk kunci KMS di akun 2 memungkinkan akun 1 untuk menggunakan kunci KMS dalam operasi kriptografi. Namun, akun 1 harus menggunakan kebijakan IAM untuk memberikan akses prinsipal ke kunci KMS.
- Kebijakan IAM di akun 1 memungkinkan `Engineering` peran untuk menggunakan kunci KMS di akun 2 untuk operasi kriptografi.
- Bob, pengguna di akun 1, memiliki izin untuk mengasumsikan peran `Engineering`.
- Bob dapat mempercayai kunci KMS ini, karena meskipun tidak ada di akunnya, kebijakan IAM di akunnya memberinya izin eksplisit untuk menggunakan kunci KMS ini.



Pertimbangkan kebijakan yang memungkinkan Bob, pengguna di akun 1, menggunakan kunci KMS di akun 2.

- Kebijakan kunci untuk kunci KMS memungkinkan akun 2 (444455556666, akun yang memiliki kunci KMS) untuk menggunakan kebijakan IAM untuk mengontrol akses ke kunci KMS. Kebijakan kunci ini juga memungkinkan akun 1 (111122223333) untuk menggunakan kunci KMS dalam operasi kriptografi (ditentukan dalam elemen pernyataan kebijakan). `Action` Namun, tidak ada seorang pun di akun 1 yang dapat menggunakan kunci KMS di akun 2 hingga akun 1 mendefinisikan kebijakan IAM yang memberikan akses kepada prinsipal ke kunci KMS.

  Dalam bagan alur, kebijakan kunci ini di akun 2 memenuhi syaratApakah kebijakan kunci memungkinkan akun pemanggil menggunakan kebijakan IAM untuk mengontrol akses ke kunci?.

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

- Kebijakan IAM di penelepon Akun AWS (akun 1, 111122223333) memberikan izin utama untuk melakukan operasi kriptografi menggunakan kunci KMS di akun 2 (444455556666). Elemen `Action` mendelegasikan akses yang sama yang diberikan kebijakan kunci di akun 2 ke akun 1 kepada perwakilan. Untuk memberikan izin ini ke peran `Engineering` dalam akun 1, kebijakan inline ini akan disematkan dalam peran `Engineering`.

  Kebijakan IAM lintas akun seperti ini hanya efektif jika kebijakan kunci untuk kunci KMS di akun 2 memberikan izin akun 1 untuk menggunakan kunci KMS. Selain itu, akun 1 hanya dapat

memberikan izin perwakilannya untuk melakukan tindakan yang diberikan kebijakan kunci kepada akun.

Dalam bagan alur, ini memenuhi syarat Apakah kebijakan IAM mengizinkan pemanggil untuk melakukan tindakan ini?.

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

- Elemen terakhir yang dibutuhkan adalah penentuan peran `Engineering` dalam akun 1. `AssumeRolePolicyDocument` dalam peran memungkinkan Bob untuk mengasumsikan peran `Engineering`.

```
{
    "Role": {
        "Arn": "arn:aws:iam::111122223333:role/Engineering",
        "CreateDate": "2019-05-16T00:09:25Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": {
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:user/bob"
                },
```

```
                "Effect": "Allow",
                "Action": "sts:AssumeRole"
            }
        },
        "Path": "/",
        "RoleName": "Engineering",
        "RoleId": "AROA4KJY2TU23Y7NK62MV"
    }
}
```

AWS KMS izin

Tabel ini dirancang untuk membantu Anda memahami AWS KMS izin sehingga Anda dapat mengontrol akses ke AWS KMS sumber daya Anda. Definisi judul kolom muncul di bawah tabel.

Anda juga dapat mempelajari tentang AWS KMS izin di tombol Tindakan, sumber daya, dan kondisi untuk AWS Key Management Service topik Referensi Otorisasi Layanan. Namun, topik tersebut tidak mencantumkan semua kunci kondisi yang dapat Anda gunakan untuk menyempurnakan setiap izin.

> **Note**
>
> Anda mungkin harus menggulir horizontal atau vertikal untuk melihat semua data di dalam tabel.

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
CancelKeyDeletion `kms:CancelKeyDeletion`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
ConnectCustomKeyStore `kms:ConnectCustomKeyStore`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
CreateAlias `kms:CreateAlias` Untuk menggunakan operasi ini, pemanggil memerlukan izin `kms:CreateAlias` pada dua sumber daya: • Alias (dalam kebijakan IAM) • Kunci KMS (dalam kebijakan utama) Untuk rincian selengkapnya, lihat Mengontrol akses ke alias.	Kebijakan IAM (untuk alias)	Tidak	Alias	Tidak ada (ketika mengontrol akses ke alias)
	Kebijakan kunci (untuk kunci KMS)	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
CreateCustomKeyStore `kms:CreateCustomKeyStore`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
CreateGrant `kms:CreateGrant`	Kebijakan kunci	Ya	Kunci KMS	Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan hibah: km: GrantConstraintType km: GranteePrincipal km: GrantIsFor AWSResource km: GrantOperations km: RetiringPrincipal Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
CreateKey `kms:CreateKey`	Kebijakan IAM	Tidak	*	km: BypassPolicyLockoutSafetyCheck km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ViaService aws:RequestTag/tag-key (kunci kondisi AWS global) aws:ResourceTag/tag-key (kunci kondisi AWS global) aws: TagKeys (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Dekripsi `kms:Decrypt`	Kebijakan kunci	Ya	Kunci KMS	Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService
DeleteAlias `kms:DeleteAlias` Untuk menggunakan operasi ini, pemanggil memerlukan izin `kms:DeleteAlias` pada dua sumber daya: • Alias (dalam kebijakan IAM) • Kunci KMS (dalam kebijakan utama) Untuk rincian selengkapnya, lihat Mengontrol akses ke alias.	Kebijakan IAM (untuk alias)	Tidak	Alias	Tidak ada (ketika mengontrol akses ke alias)
	Kebijakan kunci (untuk kunci KMS)	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
DeleteCustomKeyStore `kms:DeleteCustomKeyStore`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
DeleteImportedKeyMaterial `kms:DeleteImportedKeyMaterial`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
DescribeCustomKeyStores `kms:DescribeCustomKeyStores`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
DescribeKey `kms:DescribeKey`	Kebijakan kunci	Ya	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: RequestAlias

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
DisableKey `kms:DisableKey`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
DisableKeyRotation `kms:DisableKeyRotation`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
DisconnectCustomKeyStore `kms:DisconnectCustomKeyStore`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
EnableKey `kms:EnableKey`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
EnableKeyRotation `kms:EnableKeyRotation`	Kebijakan kunci	Tidak	Kunci KMS (hanya simetris)	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi rotasi kunci otomatis: km: RotationPeriodInDays

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Enkripsi `kms:Encrypt`	Kebijakan kunci	Ya	Kunci KMS	Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKey `kms:GenerateDataKey`	Kebijakan kunci	Ya	Kunci KMS (hanya simetris)	Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyPair `kms:GenerateDataKeyPair`	Kebijakan kunci	Ya	Kunci KMS (hanya simetris) Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	Ketentuan untuk pasangan kunci data: km: DataKeyPairSpec Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyPairWithoutPlaintext `kms:GenerateDataKeyPairWithoutPlaintext`	Kebijakan kunci	Ya	Kunci KMS (hanya simetris) Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	Ketentuan untuk pasangan kunci data: km: DataKeyPairSpec Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GenerateDataKeyWithoutPlaintext `kms:GenerateDataKeyWithoutPlaintext`	Kebijakan kunci	Ya	Kunci KMS (hanya simetris)	Kondisi untuk operasi kriptografi km: EncryptionAlgorithm km: RequestAlias Kondisi konteks enkripsi: kms:EncryptionContext: kunci-konteks km: EncryptionContextKeys Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService
GenerateMac `kms:GenerateMac`	Kebijakan kunci	Ya	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Ketentuan untuk operasi kriptografi: km: MacAlgorithm km: RequestAlias
GenerateRandom `kms:GenerateRandom`	Kebijakan IAM	N/A	*	Tidak ada

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetKeyPolicy `kms:GetKeyPolicy`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetKeyRotationStatus `kms:GetKeyRotationStatus`	Kebijakan kunci	Ya	Kunci KMS (hanya simetris)	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetParametersForImport `kms:GetParametersForImport`	Kebijakan kunci	Tidak	Kunci KMS	km: WrappingAlgorithm km: WrappingKeySpec Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
GetPublicKey `kms:GetPublicKey`	Kebijakan kunci	Ya	Kunci KMS (hanya asimetris)	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: RequestAlias

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ImportKeyMaterial `kms:ImportKeyMaterial`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: ExpirationModel km: ValidTo
ListAliases `kms:ListAliases`	Kebijakan IAM	Tidak	*	Tidak ada

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListGrants `kms:ListGrants`	Kebijakan kunci	Ya	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: GrantIsForAWSResource

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListKeyPolicies `kms:ListKeyPolicies`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListKeyRotations `kms:ListKeyRotations`	Kebijakan kunci	Tidak	Kunci KMS (hanya simetris)	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
ListKeys `kms:ListKeys`	Kebijakan IAM	Tidak	*	Tidak ada

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListResourceTags `kms:ListResourceTags`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ListRetirableGrants `kms:ListRetirableGrants`	Kebijakan IAM	Utama yang ditentukan harus berada di akun lokal, tetapi operasi mengembalikan pemberian di semua akun.	*	Tidak ada

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
PutKeyPolicy `kms:PutKeyPolicy`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: BypassPolicyLockoutSafetyCheck

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi

ReEncrypt
 	
`kms:ReEncryptFrom`
 	
`kms:ReEncryptTo`
 	
Untuk menggunakan operasi ini, penelepon memerlukan izin pada dua tombol KMS:
 	
• `kms:ReEncryptFrom` pada kunci KMS yang digunakan untuk mendekripsi
• `kms:ReEncryptTo` pada kunci KMS yang digunakan untuk mengenkripsi
	Kebijakan kunci	Ya	Kunci KMS	Kondisi untuk operasi kriptografi
				
				km: EncryptionAlgorithm
				
				km: RequestAlias
				
				Kondisi konteks enkripsi:
				
				kms:EncryptionContext: kunci-konteks
				
				km: EncryptionContextKeys
				
				Ketentuan untuk operasi kunci KMS:
				
				km: CallerAccount
				
				km: KeySpec
				
				km: KeyUsage
				
				km: KeyOrigin
				
				km: MultiRegion
				
				km: MultiRegionKeyType
				
				km: ResourceAliases
				
				aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
				km: ViaService Kondisi lain: km: ReEncryptOnSameKey
ReplicateKey kms:ReplicateKey Untuk menggunakan operasi ini, pemanggil memerlukan izin berikut: • kms:ReplicateKey pada kunci utama Multi-wilayah • kms:CreateKey dalam kebijakan IAM di Wilayah replika	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: ReplicaRegion

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
RetireGrant `kms:RetireGrant` Izin untuk mempensiunkan pemberian ditentukan terutama oleh pemberian tersebut. Kebijakan saja tidak dapat mengizinkan akses ke operasi ini. Untuk informasi selengkapnya, lihat Menghentikan dan mencabut pemberian izin.	Kebijakan IAM (Izin ini tidak efektif dalam kebijakan kunci.)	Ya	Kunci KMS	km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
RevokeGrant `kms:RevokeGrant`	Kebijakan kunci	Ya	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lain: km: GrantIsFor AWSResource

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
RotateKeyOnDemand `kms:RotateKeyOnDemand`	Kebijakan kunci	Tidak	Kunci KMS (hanya simetris)	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
ScheduleKeyDeletion `kms:ScheduleKeyDeletion`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Tanda `kms:Sign`	Kebijakan kunci	Ya	Kunci KMS (hanya asimetris)	Ketentuan untuk penandatanganan dan verifikasi: km: MessageType km: RequestAlias km: SigningAlgorithm Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
TagResource `kms:TagResource`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Ketentuan untuk menandai: aws:RequestTag/tag-key (kunci kondisi AWS global) aws: TagKeys (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
UntagResource `kms:UntagResource`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Ketentuan untuk menandai: aws:RequestTag/tag-key (kunci kondisi AWS global) aws: TagKeys (kunci kondisi AWS global)

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
UpdateAlias `kms:UpdateAlias`	Kebijakan IAM (untuk alias)	Tidak	Alias	Tidak ada (ketika mengontrol akses ke alias)
Untuk menggunakan operasi ini, pemanggil memerlukan izin `kms:UpdateAlias` pada tiga sumber daya: • Alias • Kunci KMS yang saat ini terkait • Kunci KMS yang baru terkait Untuk rincian selengkapnya, lihat Mengontrol akses ke alias.	Kebijakan kunci (untuk kunci KMS)	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService
UpdateCustomKeyStore `kms:UpdateCustomKeyStore`	Kebijakan IAM	Tidak	*	km: CallerAccount

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
UpdateKeyDescription `kms:UpdateKeyDescription`	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
UpdatePrimaryRegion `kms:UpdatePrimaryRegion` Untuk menggunakan operasi ini, pemanggil memerlukan izin `kms:UpdatePrimaryRegion` pada kedua kunci primer multi-Wilayahyang akan menjadi kunci replika dan kunci replika multi-Wilayah yang akan menjadi kunci primer.	Kebijakan kunci	Tidak	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Kondisi lainnya km: PrimaryRegion

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
Verifikasi `kms:Verify`	Kebijakan kunci	Ya	Kunci KMS (hanya asimetris)	Ketentuan untuk penandatanganan dan verifikasi: km: MessageType km: RequestAlias km: SigningAlgorithm Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService

Tindakan dan izin	Jenis kebijakan	Penggunaan lintas akun	Sumber daya (untuk kebijakan IAM)	AWS KMS kunci kondisi
VerifyMac `kms:VerifyMac`	Kebijakan kunci	Ya	Kunci KMS	Ketentuan untuk operasi kunci KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (kunci kondisi AWS global) km: ViaService Ketentuan untuk operasi kriptografi: km: MacAlgorithm km: RequestAlias

Deskripsi kolom

Kolom dalam tabel ini memberikan informasi berikut:

- Tindakan dan izin mencantumkan setiap operasi AWS KMS API dan izin yang memungkinkan operasi. Anda menentukan operasi di elemen `Action` pernyataan kebijakan.
- Jenis kebijakan menunjukkan apakah izin dapat digunakan dalam kebijakan kunci atau kebijakan IAM.

  Kebijakan kunci berarti Anda dapat menentukan izin dalam kebijakan kunci. Ketika kebijakan kunci berisi pernyataan kebijakan yang memungkinkan kebijakan IAM, Anda dapat menentukan izin dalam kebijakan IAM.

  Kebijakan IAM berarti Anda dapat menentukan izin hanya dalam kebijakan IAM.
- Penggunaan lintas akun menunjukkan operasi yang dapat dilakukan pengguna yang berwenang pada sumber daya yang berbeda Akun AWS.

  Nilai Ya berarti bahwa prinsipal dapat melakukan operasi pada sumber daya yang berbeda. Akun AWS

  Nilai Tidak berarti bahwa kepala sekolah dapat melakukan operasi hanya pada sumber daya mereka sendiri. Akun AWS

  Jika Anda memberikan utama di akun berbeda sebuah izin yang tidak dapat digunakan pada sumber lintas akun, maka izin tersebut tidak efektif. Misalnya, jika Anda memberikan prinsipal di akun lain kms: TagResource izin ke kunci KMS di akun Anda, upaya mereka untuk menandai kunci KMS di akun Anda akan gagal.
- Resources mencantumkan AWS KMS sumber daya tempat izin diterapkan. AWS KMS mendukung dua jenis sumber daya: kunci KMS dan alias. Dalam kebijakan kunci, nilai `Resource` elemen selalu`*`, yang menunjukkan kunci KMS tempat kebijakan kunci dilampirkan.

  Gunakan nilai berikut untuk mewakili AWS KMS sumber daya dalam kebijakan IAM.

  Kunci KMS

  Ketika sumber daya adalah kunci KMS, gunakan kunci ARN. Untuk bantuan, lihat the section called “Menemukan ID kunci dan kunci ARN”.

  `arn:`*`AWS_partition_name`*`:kms:`*`AWS_Region`*`:`*`AWS_account_ID`*`:key/`*`key_ID`*

  Sebagai contoh:

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

Alias

Ketika sumber daya adalah alias, gunakan ARN alias. Untuk bantuan, lihat the section called "Menemukan nama alias dan ARN alias".

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Sebagai contoh:

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

***** (tanda bintang)

Jika izin tidak berlaku untuk sumber daya tertentu (kunci KMS atau alias), gunakan tanda bintang (). *

Dalam kebijakan IAM untuk AWS KMS izin, tanda bintang dalam `Resource` elemen menunjukkan semua AWS KMS sumber daya (kunci KMS dan alias). Anda juga dapat menggunakan tanda bintang dalam `Resource` elemen ketika AWS KMS izin tidak berlaku untuk kunci atau alias KMS tertentu. Misalnya, saat mengizinkan atau menolak izin `kms:CreateKey` atau `kms:ListKeys`, Anda dapat mengatur elemen `Resource` ke `*` atau ke variasi spesifik akun, seperti `arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:*`.

- AWS KMS tombol kondisi mencantumkan tombol AWS KMS kondisi yang dapat Anda gunakan untuk mengontrol akses ke operasi. Anda menentukan syarat di elemen `Condition` kebijakan. Untuk informasi selengkapnya, lihat AWS KMS kunci kondisi. Kolom ini juga menyertakan kunci kondisi AWS global yang didukung oleh AWS KMS, tetapi tidak oleh semua AWS layanan.

Menguji izin Anda

Untuk menggunakannyaAWS KMS, Anda harus memiliki kredensi yang AWS dapat digunakan untuk mengautentikasi permintaan API Anda. Kredensialnya harus menyertakan izin untuk mengakses kunci dan alias KMS. Izin ditentukan oleh kebijakan utama, kebijakan IAM, hibah, dan kontrol akses lintas akun. Selain mengontrol akses ke kunci KMS, Anda dapat mengontrol akses ke CloudHSM Anda, dan ke toko kunci khusus Anda.

Anda dapat menentukan parameter `DryRun` API untuk memverifikasi bahwa Anda memiliki izin yang diperlukan untuk menggunakan AWS KMS kunci. Anda juga dapat menggunakan `DryRun` untuk

memverifikasi bahwa parameter permintaan dalam panggilan AWS KMS API ditentukan dengan benar.

Topik

Apa DryRun parameternya?

`DryRun`adalah parameter API opsional yang Anda tentukan untuk memverifikasi bahwa panggilan AWS KMS API akan berhasil. Gunakan `DryRun` untuk menguji panggilan API Anda, sebelum benar-benar melakukan panggilan keAWS KMS. Anda dapat memverifikasi yang berikut ini.

- Bahwa Anda memiliki izin yang diperlukan untuk menggunakan AWS KMS kunci.
- Bahwa Anda telah menentukan parameter dalam panggilan dengan benar.

AWS KMSmendukung penggunaan `DryRun` parameter dalam tindakan API tertentu:

- CreateGrant
- Dekripsi
- Enkripsi
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Tanda
- Verifikasi
- VerifyMac

Menggunakan `DryRun` parameter akan dikenakan biaya dan akan ditagih sebagai permintaan API standar. Untuk informasi lebih lanjut tentang harga AWS KMS, lihat Harga AWS Key Management Service.

Semua permintaan API yang menggunakan `DryRun` parameter berlaku untuk kuota permintaan API dan dapat menghasilkan pengecualian pembatasan jika Anda melebihi kuota permintaan API. Misalnya, memanggil Dekripsi dengan `DryRun` atau tanpa `DryRun` hitungan terhadap kuota operasi kriptografi yang sama. Lihat Permintaan pelambatan AWS KMS untuk mempelajari selengkapnya.

Setiap panggilan ke operasi AWS KMS API ditangkap sebagai peristiwa dan direkam dalam AWS CloudTrail log. Output dari setiap operasi yang menentukan `DryRun` parameter muncul di CloudTrail log Anda. Untuk informasi selengkapnya, lihat Logging panggilan AWS KMS API dengan AWS CloudTrail.

Menentukan DryRun dengan API

Untuk menggunakan`DryRun`, tentukan `–dry-run` parameter dalam AWS CLI perintah dan panggilan AWS KMS API yang mendukung parameter. Ketika Anda melakukannya, AWS KMS akan memverifikasi apakah panggilan Anda akan berhasil. AWS KMSpanggilan yang digunakan `DryRun` akan selalu gagal dan mengembalikan pesan dengan informasi tentang alasan mengapa panggilan gagal. Pesan dapat mencakup pengecualian berikut:

- `DryRunOperationException`- Permintaan akan berhasil jika `DryRun` tidak ditentukan.
- `ValidationException`- Permintaan gagal menentukan parameter API yang salah.
- `AccessDeniedException`- Anda tidak memiliki izin untuk melakukan tindakan API yang ditentukan pada sumber daya KMS.

Misalnya, perintah berikut menggunakan CreateGrantoperasi dan membuat hibah yang memungkinkan pengguna yang berwenang untuk mengambil `keyUserRole` peran untuk memanggil operasi Dekripsi pada kunci KMS simetris tertentu. `DryRun`Parameter ditentukan.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

Kunci tujuan khusus

AWS Key Management Service(AWS KMS) mendukung beberapa jenis kunci untuk penggunaan yang berbeda.

Saat Anda membuatAWS KMS key, secara default, Anda mendapatkan kunci KMS enkripsi simetris. DalamAWS KMS, kunci KMS enkripsi simetris mewakili kunci AES-GCM 256-bit yang digunakan untuk enkripsi dan dekripsi, kecuali di Wilayah China, di mana ia mewakili simetris kunci simetris 128-bit yang menggunakan enkripsi SM4. Bahan kunci simetris tidak pernah dibiarkan tidak AWS KMS terenkripsi. Kecuali tugas Anda secara eksplisit memerlukan enkripsi asimetris atau kunci HMAC, kunci KMS enkripsi simetris, yang tidak pernah dibiarkan AWS KMS tidak terenkripsi, adalah pilihan yang baik. Juga, AWSlayanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris.

Anda dapat menggunakan kunci KMS enkripsi simetris AWS KMS untuk mengenkripsi, mendekripsi, dan mengenkripsi ulang data, menghasilkan kunci data dan pasangan kunci data, dan menghasilkan string byte acak. Anda dapat mengimpor materi kunci Anda sendiri ke kunci KMS enkripsi simetris dan membuat kunci KMS enkripsi simetris di toko kunci khusus. Untuk tabel yang membandingkan operasi yang dapat Anda lakukan pada tombol KMS simetris dan asimetris, lihat. Referensi tipe kunci

AWS KMSjuga mendukung jenis kunci KMS tujuan khusus berikut:

- Kunci RSA asimetris untuk kriptografi kunci publik
- Kunci RSA dan ECC asimetris untuk penandatanganan dan verifikasi
- Kunci SM2 asimetris (khusus Wilayah China) untuk kriptografi kunci publik atau penandatanganan dan verifikasi
- Kunci HMAC untuk menghasilkan dan memverifikasi kode otentikasi pesan berbasis hash
- Tombol Multi-Region (simetris dan asimetris) yang berfungsi seperti salinan dari kunci yang sama dalam berbagai Wilayah AWS
- Kunci dengan bahan kunci impor yang Anda berikan
- Kunci di penyimpanan kunci khusus yang didukung oleh AWS CloudHSM cluster atau pengelola kunci eksternal di luarAWS.

Memilih tipe kunci KMS

AWS KMSmendukung beberapa jenis kunci KMS: kunci enkripsi simetris, kunci HMAC simetris, kunci enkripsi asimetris, dan kunci penandatanganan asimetris.

Kunci KMS berbeda karena mengandung bahan kunci kriptografi yang berbeda.

- Kunci KMS enkripsi simetris: Merupakan kunci enkripsi AES-GCM 256-bit tunggal, kecuali di Wilayah China, di mana ia mewakili kunci enkripsi SM4 128-bit. Bahan kunci simetris tidak pernah dibiarkan tidak AWS KMS terenkripsi. Untuk menggunakan kunci KMS enkripsi simetris Anda, Anda harus menelepon. AWS KMS

  Kunci enkripsi simetris, yang merupakan kunci KMS default, ideal untuk sebagian besar penggunaan. Jika Anda memerlukan kunci KMS untuk melindungi data AndaLayanan AWS, gunakan kunci enkripsi simetris kecuali Anda diperintahkan untuk menggunakan jenis kunci lain.
- Kunci KMS asimetris: Merupakan kunci publik dan private key pair terkait matematis yang dapat Anda gunakan untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi, tetapi tidak keduanya. Kunci privat tidak pernah membiarkan AWS KMS tidak terenkripsi. Anda dapat menggunakan kunci publik dalam AWS KMS dengan memanggil operasi AWS KMS API, atau mengunduh kunci publik dan menggunakannya di luar AWS KMS.
- Kunci KMS HMAC (simetris): Merupakan kunci simetris dengan panjang yang bervariasi yang digunakan untuk menghasilkan dan memverifikasi kode otentikasi pesan berbasis hash. Materi kunci dalam kunci HMAC KMS tidak pernah dibiarkan tidak terenkripsi. AWS KMS Untuk menggunakan kunci HMAC KMS Anda, Anda harus menelepon. AWS KMS

Jenis kunci KMS yang Anda buat sangat tergantung pada bagaimana Anda berencana untuk menggunakan kunci KMS, persyaratan keamanan Anda, dan persyaratan otorisasi Anda. Saat membuat kunci KMS Anda, ingatlah bahwa konfigurasi kriptografi kunci KMS, termasuk spesifikasi kunci dan penggunaan kuncinya, ditetapkan saat Anda membuat kunci KMS dan tidak dapat diubah.

Gunakan panduan berikut untuk menentukan jenis kunci KMS yang Anda butuhkan berdasarkan kasus penggunaan Anda.

Mengenkripsi dan mendekripsi data

Gunakan kunci KMS simetris untuk sebagian besar kasus penggunaan yang memerlukan enkripsi dan dekripsi data. Algoritme enkripsi simetris yang digunakan AWS KMS berfungsi dengan cepat, efisien, dan menjamin kerahasiaan dan keaslian data. Agoritme ini mendukung enkripsi

yang diautentikasi dengan data tambahan yang diautentikasi (AAD), yang didefinisikan sebagai Konteks enkripsi. Jenis kunci KMS ini mengharuskan pengirim dan penerima data terenkripsi untuk memiliki kredensil yang valid AWS untuk dipanggil. AWS KMS

Jika kasus penggunaan Anda memerlukan enkripsi di luar AWS oleh pengguna yang tidak dapat meneleponAWS KMS, kunci KMS asimetris adalah pilihan yang baik. Anda dapat mendistribusikan kunci publik dari kunci KMS asimetris untuk memungkinkan pengguna ini mengenkripsi data. Dan aplikasi Anda yang perlu mendekripsi data tersebut dapat menggunakan kunci pribadi dari kunci KMS asimetris di dalamnya. AWS KMS

Menandatangani pesan dan memverifikasi tanda tangan

Untuk menandatangani pesan dan memverifikasi tanda tangan, Anda harus menggunakan kunci KMS asimetris. Anda dapat menggunakan kunci KMS dengan spesifikasi kunci yang mewakili key pair RSA, elliptic curve (ECC) key pair, atau SM2 key pair (China Regions only). Spesifikasi kunci yang Anda pilih ditentukan oleh algoritme penandatanganan yang ingin Anda gunakan. Algoritma penandatanganan ECDSA yang didukung oleh pasangan kunci ECC direkomendasikan melalui algoritma penandatanganan RSA. Namun, Anda mungkin perlu menggunakan spesifikasi kunci tertentu dan algoritma penandatanganan untuk mendukung pengguna yang memverifikasi tanda tangan di luar. AWS

Melakukan enkripsi kunci publik

Untuk melakukan enkripsi kunci publik, Anda harus menggunakan kunci KMS asimetris dengan spesifikasi kunci RSA atau spesifikasi kunci SM2 (khusus Wilayah China). Untuk mengenkripsi data AWS KMS dengan kunci publik dari key pair KMS, gunakan operasi Encrypt. Anda juga dapat mengunduh kunci publik dan membagikannya dengan pihak yang perlu mengenkripsi data di luar AWS KMS.

Saat Anda mengunduh kunci publik dari kunci KMS asimetris, Anda dapat menggunakannya di luar. AWS KMS Tapi itu tidak lagi tunduk pada kontrol keamanan yang melindungi kunci KMS. AWS KMS Misalnya, Anda tidak dapat menggunakan kebijakan atau hibah kunci AWS KMS untuk mengontrol penggunaan kunci publik. Anda juga tidak dapat mengontrol apakah kunci hanya digunakan untuk enkripsi dan dekripsi menggunakan algoritma enkripsi yang mendukung. AWS KMS Untuk detail selengkapnya, lihat Pertimbangan Khusus untuk Mengunduh Kunci Publik.

Untuk mendekripsi data yang dienkripsi dengan kunci publik di luar AWS KMS, panggil operasi Dekripsi. `Decrypt`Operasi gagal jika data dienkripsi di bawah kunci publik dari kunci KMS dengan penggunaan kunci. `SIGN_VERIFY` Ini juga akan gagal jika dienkripsi dengan menggunakan algoritma yang AWS KMS tidak mendukung spesifikasi kunci yang Anda pilih. Untuk informasi

selengkapnya tentang spesifikasi utama dan algoritme yang didukung, lihat Spesifikasi kunci asimetris.

Untuk menghindari kesalahan ini, siapa pun yang menggunakan kunci publik di luar AWS KMS harus menyimpan konfigurasi kunci. AWS KMSKonsol dan GetPublicKeyrespons memberikan informasi yang harus Anda sertakan saat Anda membagikan kunci publik.

Hasilkan dan verifikasi kode HMAC

Untuk membuat dan memverifikasi kode otentikasi pesan berbasis hash, gunakan kunci HMAC KMS. Saat Anda membuat kunci HMACAWS KMS, AWS KMS membuat dan melindungi materi kunci Anda dan memastikan bahwa Anda menggunakan algoritma MAC yang benar untuk kunci Anda. Kode HMAC juga dapat digunakan sebagai nomor pseudo-acak, dan dalam skenario tertentu untuk penandatanganan dan tokenisasi simetris.

Kunci HMAC KMS adalah kunci simetris. Saat membuat kunci HMAC KMS di AWS KMS konsol, pilih jenis `Symmetric` kunci.

Gunakan dengan AWS layanan

Untuk membuat kunci KMS untuk digunakan dengan AWSlayanan yang terintegrasi dengan AWS KMS, lihat dokumentasi untuk layanan tersebut. AWSlayanan yang mengenkripsi data Anda memerlukan kunci KMS enkripsi simetris. .

Selain pertimbangan ini, operasi kriptografi pada kunci KMS dengan spesifikasi kunci yang berbeda memiliki harga yang berbeda dan kuota permintaan yang berbeda. Untuk informasi selengkapnya tentang harga AWS KMS, lihat Harga AWS Key Management Service. Untuk informasi selengkapnya tentang kuota permintaan, lihat Kuota permintaan.

Memilih penggunaan kunci

Penggunaan kunci KMS menentukan apakah kunci KMS digunakan untuk enkripsi dan dekripsi, atau menandatangani dan memverifikasi tanda tangan, atau membuat dan memverifikasi tag HMAC. Setiap kunci KMS hanya memiliki satu penggunaan kunci. Menggunakan kunci KMS untuk lebih dari satu jenis operasi membuat produk dari semua operasi lebih rentan terhadap serangan.

Seperti yang ditunjukkan pada tabel berikut, kunci KMS enkripsi simetris hanya dapat digunakan untuk enkripsi dan dekripsi. Kunci HMAC KMS hanya dapat digunakan untuk menghasilkan dan memverifikasi kode HMAC. Kunci KMS kurva elips (ECC) hanya dapat digunakan untuk

penandatanganan dan verifikasi. Anda perlu membuat keputusan penggunaan kunci hanya untuk kunci RSA KMS.

Penggunaan kunci yang valid untuk tipe kunci KMS

Jenis kunci KMS	Enkripsi dan dekripsi ENCRYPT_DECRYPT	Tanda tangan dan verifikasi SIGN_VERIFY	Hasilkan dan verifikasi MAC GENERATE_VERIFY_MAC
Kunci KMS enkripsi simetris	✓	⊗	⊗
Kunci HMAC KMS (simetris)	⊗	⊗	✓
Kunci KMS asimetris dengan pasangan kunci RSA	✓	✓	⊗
Kunci KMS asimetris dengan pasangan kunci ECC	⊗	✓	⊗
Kunci KMS asimetris dengan pasangan kunci SM2 (hanya Wilayah China)	✓	✓	⊗

Di AWS KMS konsol, pertama-tama Anda memilih jenis kunci (simetris atau asimetris) dan kemudian penggunaan kunci. Jenis kunci yang Anda pilih menentukan opsi penggunaan kunci mana yang ditampilkan. Penggunaan kunci yang Anda pilih menentukan spesifikasi kunci mana, jika ada, yang ditampilkan.

Untuk memilih penggunaan kunci dalam konsol AWS KMS:

- Untuk kunci KMS enkripsi simetris (default), pilih Enkripsi dan dekripsi.
- Untuk kunci HMAC KMS, pilih Hasilkan dan verifikasi MAC.

- Untuk kunci KMS asimetris dengan bahan kunci kurva elips (ECC), pilih Tanda dan verifikasi.
- Untuk kunci KMS asimetris dengan bahan kunci RSA, pilih Enkripsi dan dekripsi atau Tandatangani dan verifikasi.
- Untuk kunci KMS asimetris dengan bahan kunci SM2, pilih Enkripsi dan dekripsi atau Tandatangani dan verifikasi. Spesifikasi kunci SM2 hanya tersedia di Wilayah China.

Untuk mengizinkan prinsipal membuat kunci KMS hanya untuk penggunaan kunci tertentu, gunakan kunci kondisi kms:. KeyUsage Anda juga dapat menggunakan kunci `kms:KeyUsage` kondisi untuk mengizinkan prinsipal memanggil operasi API untuk kunci KMS berdasarkan penggunaan kuncinya. Misalnya, Anda dapat mengizinkan izin untuk menonaktifkan kunci KMS hanya jika penggunaan kuncinya adalah SIGN_VERIFY.

Memilih spesifikasi kunci

Saat Anda membuat kunci KMS asimetris atau kunci HMAC KMS, Anda memilih spesifikasi kuncinya. Spesifikasi kunci, yang merupakan properti dari setiapAWS KMS key, mewakili konfigurasi kriptografi kunci KMS Anda. Anda memilih spesifikasi kunci saat Anda membuat kunci KMS, dan Anda tidak dapat mengubahnya. Jika Anda memilih spesifikasi kunci yang salah, hapus kunci KMS, dan buat yang baru.

> **Note**
>
> Spesifikasi kunci untuk kunci KMS dikenal sebagai “spesifikasi kunci master pelanggan.” `CustomerMasterKeySpec`Parameter CreateKeyoperasi tidak digunakan lagi. Sebagai gantinya, gunakan `KeySpec` parameternya. Respons dari `CreateKey` dan DescribeKeyoperasi termasuk `CustomerMasterKeySpec` anggota `KeySpec` dan dengan nilai yang sama.

Spesifikasi kunci menentukan apakah kunci KMS simetris atau asimetris, jenis bahan kunci dalam kunci KMS, dan algoritma enkripsi, algoritma penandatanganan, atau algoritma kode otentikasi pesan (MAC) yang mendukung kunci KMS. AWS KMS Spesifikasi kunci yang Anda pilih biasanya ditentukan oleh kasus penggunaan dan persyaratan peraturan. Namun, operasi kriptografi pada kunci KMS dengan spesifikasi kunci yang berbeda dihargai secara berbeda dan tunduk pada kuota yang berbeda. Untuk detail harga, lihat AWS Key Management ServiceHarga. Untuk informasi selengkapnya tentang kuota permintaan, lihat Kuota permintaan.

Untuk menentukan spesifikasi utama yang diizinkan oleh prinsipal di akun Anda untuk kunci KMS, gunakan kunci kondisi kms:. KeySpec

AWS KMSmendukung spesifikasi kunci berikut untuk kunci KMS:

Spesifikasi kunci enkripsi simetris (default)

- SYMMETRIC_DEFAULT

Spesifikasi kunci HMAC

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

Spesifikasi kunci RSA (enkripsi dan dekripsi -atau- penandatanganan dan verifikasi)

- RSA_2048
- RSA_3072
- RSA_4096

Spesifikasi kunci kurva eliptik

- Pasangan kunci kurva elips (penandatanganan dan verifikasi) rekomendasi NIST Asimetris
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Pasangan kunci kurva elips asimetri lain (penandatanganan dan verifikasi)
  - ECC_SECG_P256K1 (secp256k1), biasa digunakan untuk mata uang kripto.

Spesifikasi kunci SM2 (enkripsi dan dekripsi -atau- penandatanganan dan verifikasi)

- SM2 (Hanya Wilayah China)

Kunci asimetris di AWS KMS

AWS KMSmendukung kunci KMS asimetris yang mewakili RSA terkait matematis, kurva eliptik (ECC), atau SM2 (China Regions only) public dan private key pair. Pasangan kunci ini dihasilkan dalam modul keamanan AWS KMS perangkat keras yang disertifikasi di bawah Program Validasi Modul Kriptografi FIPS 140-2, kecuali di Wilayah China (Beijing) dan China (Ningxia). Kunci pribadi

tidak pernah meninggalkan AWS KMS HSM tidak terenkripsi. Anda dapat mengunduh kunci publik untuk distribusi dan digunakan di luarAWS. Anda dapat membuat kunci KMS asimetris untuk enkripsi dan dekripsi, atau penandatanganan dan verifikasi, tetapi tidak keduanya.

Anda dapat membuat dan mengelola kunci KMS asimetris di AndaAkun AWS, termasuk menyetel kebijakan utama, kebijakanIAM, dan hibah yang mengontrol akses ke kunci, mengaktifkan dan menonaktifkan kunci KMS, membuat tag dan alias, dan menghapus kunci KMS. Anda dapat mengaudit semua operasi yang menggunakan atau mengelola kunci KMS asimetris Anda AWS dalam AWS CloudTrail log.

AWS KMSjuga menyediakan pasangan kunci data asimetris yang dirancang untuk digunakan untuk kriptografi sisi klien di luar. AWS KMS Kunci pribadi dalam asymmetric data key pair dilindungi oleh kunci KMS enkripsi simetris. AWS KMS

Topik ini menjelaskan bagaimana kunci KMS asimetris bekerja, bagaimana mereka berbeda dari kunci KMS lainnya dan bagaimana memutuskan jenis kunci KMS yang Anda butuhkan untuk melindungi data Anda. Ini juga menjelaskan bagaimana pasangan kunci data asimetris bekerja dan bagaimana menggunakannya di luar. AWS KMS

Daerah

Kunci KMS asimetris dan pasangan kunci data asimetris didukung di semua Wilayah AWS yang mendukung. AWS KMS

Pelajari selengkapnya

- Untuk membuat kunci KMS asimetris, lihat. Membuat tombol KMS asimetris Untuk membuat kunci KMS enkripsi simetris, lihat. Membuat kunci
- Untuk membuat kunci KMS asimetris Multi-wilayah, lihat. Membuat kunci multi-Wilayah
- Untuk mengetahui apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris
- Untuk tabel yang membandingkan operasi AWS KMS API yang berlaku untuk setiap jenis kunci KMS, lihat. the section called “Referensi tipe kunci”
- Untuk mengontrol akses ke spesifikasi utama, penggunaan kunci, algoritme enkripsi, dan algoritme penandatanganan yang dapat digunakan oleh prinsipal di akun Anda untuk kunci KMS dan kunci data, lihat. the section called “AWS KMS kunci kondisi”
- Untuk mempelajari kuota permintaan yang berlaku untuk berbagai jenis kunci KMS, lihat. the section called “Kuota permintaan”

- Untuk mempelajari cara menandatangani pesan dan memverifikasi tanda tangan dengan kunci KMS asimetris, lihat Penandatanganan digital dengan fitur kunci asimetris baru di Blog Keamanan. AWS KMS AWS

Topik

Kunci Asymmetric KMS

Anda dapat membuat kunci KMS asimetris di. AWS KMS Kunci KMS asimetris mewakili kunci publik yang terkait secara matematis dan private key pair. Anda dapat memberikan kunci publik kepada siapa pun, meskipun mereka tidak dipercaya, tetapi kunci pribadi harus dirahasiakan.

Dalam kunci KMS asimetris, kunci pribadi dibuat AWS KMS dan tidak pernah meninggalkan AWS KMS yang tidak terenkripsi. Untuk menggunakan kunci privat, Anda harus memanggil AWS KMS. Anda dapat menggunakan kunci publik dalam AWS KMS dengan memanggil operasi AWS KMS API. Atau, Anda dapat mengunduh kunci publik dan menggunakannya di luar AWS KMS.

Jika kasus penggunaan Anda memerlukan enkripsi di luar AWS oleh pengguna yang tidak dapat meneleponAWS KMS, kunci KMS asimetris adalah pilihan yang baik. Namun, jika Anda membuat kunci KMS untuk mengenkripsi data yang Anda simpan atau kelola dalam suatu AWS layanan, gunakan kunci KMS enkripsi simetris. AWSlayanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris.

AWS KMSmendukung tiga jenis kunci KMS asimetris.

- Kunci KMS RSA: Kunci KMS dengan key pair RSA untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi (tetapi tidak keduanya). AWS KMSmendukung beberapa panjang kunci untuk persyaratan keamanan yang berbeda.
- Kunci KMS Elliptic Curve (ECC): Kunci KMS dengan elliptic curve key pair untuk penandatanganan dan verifikasi. AWS KMSmendukung beberapa kurva yang umum digunakan.

- Kunci SM2 KMS (hanya Wilayah China): Kunci KMS dengan key pair SM2 untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi (tetapi tidak keduanya).

Untuk bantuan memilih konfigurasi kunci asimetris Anda, lihatMemilih tipe kunci KMS. Untuk detail teknis tentang enkripsi dan algoritma penandatanganan yang AWS KMS mendukung kunci RSA KMS, lihat spesifikasi kunci RSA. Untuk detail teknis tentang algoritma penandatanganan yang AWS KMS mendukung kunci ECC KMS, lihat Spesifikasi kunci kurva elips. Untuk detail teknis tentang enkripsi dan algoritma penandatanganan yang AWS KMS mendukung kunci SM2 KMS (khusus Wilayah China), lihat spesifikasi kunci SM2.

Untuk tabel yang membandingkan operasi yang dapat Anda lakukan pada tombol KMS simetris dan asimetris, lihat Membandingkan kunci KMS Simetris dan Asimetris. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Daerah

Kunci KMS asimetris dan pasangan kunci data asimetris didukung di semua Wilayah AWS yang mendukung. AWS KMS

Membuat tombol KMS asimetris

Anda dapat membuat kunci KMS asimetris di AWS KMS konsol, dengan menggunakan CreateKeyAPI, atau dengan menggunakan template. AWS CloudFormation Kunci KMS asimetris mewakili key pair publik dan pribadi yang dapat digunakan untuk enkripsi atau penandatanganan. Kunci privat tetap berada dalam AWS KMS. Untuk mengunduh kunci publik untuk digunakan di luar AWS KMS, lihat Mengunduh kunci publik.

Saat membuat kunci KMS untuk mengenkripsi data yang Anda simpan atau kelola dalam suatu AWS layanan, gunakan kunci KMS enkripsi simetris. AWSlayanan yang terintegrasi dengan AWS KMS tidak mendukung kunci KMS asimetris. Untuk bantuan memutuskan apakah akan membuat kunci KMS simetris atau asimetris, lihat. Memilih tipe kunci KMS

Untuk informasi tentang izin yang diperlukan untuk membuat kunci KMS, lihat. Izin untuk membuat kunci KMS

Topik

- Membuat tombol KMS asimetris (konsol)
- Membuat kunci KMS asimetris (API) AWS KMS

Membuat tombol KMS asimetris (konsol)

Anda dapat menggunakan tombol AWS Management Console untuk membuat asimetris AWS KMS keys (tombol KMS). Setiap kunci KMS asimetris mewakili public dan private key pair.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Untuk membuat kunci KMS asimetris, dalam Key type, pilih Asymmetric.

   Untuk informasi tentang cara membuat kunci KMS enkripsi simetris di AWS KMS konsol, lihat. Membuat kunci KMS enkripsi simetris (konsol)
6. Untuk membuat kunci KMS asimetris untuk enkripsi kunci publik, dalam penggunaan Kunci, pilih Enkripsi dan dekripsi. Atau, untuk membuat kunci KMS asimetris untuk menandatangani pesan dan memverifikasi tanda tangan, dalam Penggunaan kunci, pilih Masuk dan verifikasi.

   Untuk bantuan memilih nilai penggunaan kunci, lihat Memilih penggunaan kunci.
7. Pilih spesifikasi (Spesifikasi kunci) untuk kunci KMS asimetris Anda.

   Seringkali spesifikasi kunci yang Anda pilih ditentukan oleh persyaratan peraturan, keamanan, atau bisnis. Mungkin juga dipengaruhi oleh ukuran pesan yang harus Anda enkripsi atau tanda tangani. Secara umum, kunci enkripsi yang lebih panjang lebih tahan terhadap serangan brutal.

   Untuk bantuan memilih spesifikasi kunci, lihat Memilih spesifikasi kunci.
8. Pilih Selanjutnya.
9. Ketik alias untuk kunci KMS. Nama alias tidak dapat dimulai dengan **`aws/`**. **`aws/`**Awalan dicadangkan oleh Amazon Web Services untuk mewakili Kunci yang dikelola AWS di akun Anda.

Alias adalah nama ramah yang dapat Anda gunakan untuk mengidentifikasi kunci KMS di konsol dan di beberapa AWS KMS API. Kami menyarankan Anda memilih alias yang menunjukkan jenis data yang Anda rencanakan untuk dilindungi atau aplikasi yang Anda rencanakan untuk digunakan dengan kunci KMS.

Alias diperlukan saat Anda membuat kunci KMS di file. AWS Management Console Anda tidak dapat menentukan alias ketika Anda menggunakan CreateKeyoperasi, tetapi Anda dapat menggunakan konsol atau CreateAliasoperasi untuk membuat alias untuk kunci KMS yang ada. Untuk detailnya, lihat Menggunakan alias.

10. (Opsional) Ketik deskripsi untuk kunci KMS.

    Masukkan deskripsi yang menjelaskan jenis data yang Anda rencanakan untuk dilindungi atau aplikasi yang Anda rencanakan untuk digunakan dengan kunci KMS.

    Anda dapat menambahkan deskripsi sekarang atau memperbaruinya kapan saja kecuali status kunci adalah `Pending Deletion` atau`Pending Replica Deletion`. Untuk menambah, mengubah, atau menghapus deskripsi kunci terkelola pelanggan yang ada, edit deskripsi di AWS Management Console atau gunakan UpdateKeyDescriptionoperasi.

11. (Opsional) Ketik kunci tanda dan nilai tanda opsional. Untuk menambahkan lebih dari satu tag ke tombol KMS, pilih Tambah tag.

    Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS

12. Pilih Berikutnya.

13. Pilih pengguna IAM dan peran yang dapat mengelola kunci KMS.

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini.
> Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk memberikan izin kepada prinsipal lain untuk mengelola kunci KMS. Untuk detailnya, lihat the section called “Kebijakan kunci default”.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi

sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

14. (Opsional) Untuk mencegah pengguna dan peran IAM yang dipilih menghapus kunci KMS ini, di bagian Penghapusan kunci di bagian bawah halaman, kosongkan kotak centang Izinkan administrator kunci untuk menghapus kunci ini.
15. Pilih Berikutnya.
16. Pilih pengguna IAM dan peran yang dapat menggunakan kunci KMS untuk operasi kriptografi.

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini. Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk memberikan izin kepada prinsipal lain untuk menggunakan kunci KMS dalam operasi kriptografi. Untuk detailnya, lihat the section called “Kebijakan kunci default”. Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

17. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, dalam bagian Lainnya Akun AWS di bawah halaman, pilih Tambahkan Akun AWS lain dan masukkan nomor identifikasi Akun AWS akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Untuk mengizinkan prinsipal di akun eksternal menggunakan kunci KMS, administrator akun eksternal harus membuat kebijakan IAM yang memberikan izin ini. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

18. Pilih Selanjutnya.
19. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
20. Pilih Selesai untuk membuat kunci KMS.

Membuat kunci KMS asimetris (API) AWS KMS

Anda dapat menggunakan CreateKeyoperasi untuk membuat asimetrisAWS KMS key. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Saat Anda membuat kunci KMS asimetris, Anda harus menentukan `KeySpec` parameter, yang menentukan jenis kunci yang Anda buat. Juga, Anda harus menentukan nilai `KeyUsage` ENCRYPT_DECRYPT atau SIGN_VERIFY. Anda tidak dapat mengubah properti ini setelah kunci KMS dibuat.

`CreateKey`Operasi tidak memungkinkan Anda menentukan alias, tetapi Anda dapat menggunakan CreateAliasoperasi untuk membuat alias untuk kunci KMS baru Anda.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Contoh berikut menggunakan `CreateKey` operasi untuk membuat kunci KMS asimetris dari kunci RSA 4096-bit yang dirancang untuk enkripsi kunci publik.

```
$ 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
    }
}
```

Contoh perintah berikut membuat kunci KMS asimetris yang mewakili sepasang kunci ECDSA yang digunakan untuk penandatanganan dan verifikasi. Anda tidak dapat membuat pasangan kunci kurva elips untuk enkripsi dan dekripsi.

```
$ 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"
    }
}
```

Mengunduh kunci publik

Anda dapat melihat, menyalin, dan mengunduh kunci publik dari key pair KMS asimetris dengan menggunakan AWS Management Console atau API. AWS KMS Anda harus memiliki `kms:GetPublicKey` izin pada kunci KMS asimetris.

Setiap key pair KMS asimetris terdiri dari kunci pribadi yang tidak pernah meninggalkan AWS KMS unenkripsi dan kunci publik yang dapat Anda unduh dan bagikan.

Anda dapat berbagi kunci publik untuk membiarkan lainnya mengenkripsi data di luar AWS KMS bahwa Anda dapat mendekripsi hanya dengan kunci privat Anda. Atau, untuk mengizinkan orang lain memverifikasi tanda tangan digital di luar AWS KMS yang telah Anda hasilkan dengan kunci privat Anda.

Saat Anda menggunakan kunci publik di kunci KMS asimetris di dalamnyaAWS KMS, Anda mendapat manfaat dari otentikasi, otorisasi, dan pencatatan yang merupakan bagian dari setiap operasi. AWS KMS Anda juga mengurangi risiko mengenkripsi data yang tidak dapat didekripsi. Fitur-fitur ini tidak efektif di luar AWS KMS. Untuk rincian selengkapnya, lihat Pertimbangan khusus untuk mengunduh kunci publik.

> **Tip**
>
> Mencari kunci data atau kunci SSH? Topik ini menjelaskan cara mengelola kunci asimetrisAWS Key Management Service, di mana kunci pribadi tidak dapat diekspor. Untuk pasangan kunci data yang dapat diekspor di mana kunci pribadi dilindungi oleh kunci KMS enkripsi simetris, lihat. GenerateDataKeyPair Untuk bantuan dalam mengunduh kunci publik yang terkait dengan instans Amazon EC2, lihat Mengambil kunci publik di Panduan Pengguna Amazon EC2 untuk Instans Linux dan Panduan Pengguna Amazon EC2 untuk Instans Windows.

Topik

Pertimbangan khusus untuk mengunduh kunci publik

Untuk melindungi kunci KMS Anda, AWS KMS berikan kontrol akses, enkripsi yang diautentikasi, dan log terperinci dari setiap operasi. AWS KMSjuga memungkinkan Anda untuk mencegah penggunaan kunci KMS, sementara atau permanen. Akhirnya, operasi AWS KMS dirancang untuk meminimalkan risiko mengenkripsi data yang tidak dapat didekripsi. Fitur ini tidak tersedia bila Anda menggunakan kunci publik yang diunduh di luar AWS KMS.

Otorisasi

Kebijakan utama dan kebijakan IAM yang mengontrol akses ke kunci KMS di dalamnya tidak AWS KMS berpengaruh pada operasi yang dilakukan di luar. AWS Setiap pengguna yang bisa mendapatkan kunci publik dapat menggunakannya di luar AWS KMS meskipun mereka tidak memiliki izin untuk mengenkripsi data atau memverifikasi tanda tangan dengan kunci KMS.

Pembatasan penggunaan kunci

Pembatasan penggunaan kunci tidak efektif di luar AWS KMS. Jika Anda memanggil operasi Enkripsi dengan kunci KMS yang memiliki `KeyUsage` of`SIGN_VERIFY`, AWS KMS operasi gagal. Tetapi jika Anda mengenkripsi data di luar AWS KMS dengan kunci publik dari kunci KMS dengan dari`SIGN_VERIFY`, data tidak dapat didekripsi. `KeyUsage`

Pembatasan algoritme

Pembatasan algoritme enkripsi dan penandatanganan yang didukung AWS KMS tidak efektif di luar AWS KMS. Jika Anda mengenkripsi data dengan kunci publik dari kunci KMS di luarAWS KMS, dan menggunakan algoritma enkripsi yang tidak mendukung, data AWS KMS tidak dapat didekripsi.

Menonaktifkan dan menghapus kunci KMS

Tindakan yang dapat Anda ambil untuk mencegah penggunaan kunci KMS dalam operasi kriptografi di dalam AWS KMS tidak mencegah siapa pun menggunakan kunci publik di luar. AWS KMS Misalnya, menonaktifkan kunci KMS, menjadwalkan penghapusan kunci KMS, menghapus kunci KMS, atau menghapus materi kunci dari kunci KMS tidak berpengaruh pada kunci publik di luar. AWS KMS Jika Anda menghapus kunci KMS asimetris atau menghapus atau kehilangan materi kuncinya, data yang Anda enkripsi dengan kunci publik di luar tidak dapat dipulihkan. AWS KMS

Pencatatan log

Log AWS CloudTrail yang merekam setiap operasi AWS KMS, termasuk permintaan, respons, tanggal, waktu, dan pengguna terotorisasi, tidak mencatat penggunaan kunci publik di luar AWS KMS.

Verifikasi offline dengan pasangan kunci SM2 (hanya Wilayah China)

Untuk memverifikasi tanda tangan di luar AWS KMS dengan kunci publik SM2, Anda harus menentukan ID pembeda. Secara default, AWS KMS digunakan `1234567812345678` sebagai ID pembeda. Untuk informasi selengkapnya, lihat Verifikasi offline dengan pasangan kunci SM2 (khusus Wilayah China).

Mengunduh kunci publik (konsol)

Anda dapat menggunakan AWS Management Console untuk melihat, menyalin, dan mengunduh kunci publik dari kunci KMS asimetris di kunci Anda. Akun AWS Untuk mengunduh kunci publik dari kunci KMS asimetris secara berbedaAkun AWS, gunakan API. AWS KMS

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih alias atau ID kunci dari kunci KMS asimetris.
5. Pilih tab Konfigurasi kriptografi. Catat nilai-nilai dari bidang Spesifikasi kunci, Penggunaan kunci, dan Algoritme enkripsi atau Penandatanganan Algoritme. Anda harus menggunakan nilai-nilai ini untuk menggunakan kunci publik di luar dari AWS KMS. Pastikan untuk membagikan informasi ini ketika Anda membagikan kunci publik.
6. Pilih tab Kunci publik.
7. Untuk menyalin kunci publik ke clipboard Anda, pilih Salin. Untuk mengunduh kunci publik ke file, pilih Unduh.

Mengunduh kunci publik (API AWS KMS)

GetPublicKeyOperasi mengembalikan kunci publik dalam kunci KMS asimetris. Ini juga mengembalikan informasi penting yang Anda butuhkan untuk menggunakan kunci publik dengan benar di luar AWS KMS, termasuk penggunaan kunci dan algoritme enkripsi. Pastikan untuk menyimpan nilai-nilai ini dan membagikannya setiap kali Anda berbagi kunci publik.

Contoh-contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Untuk menentukan kunci KMS, gunakan ID kunci, kunciARN, nama alias, atau alias ARN.
Bila menggunakan nama alias, beri prefiks dengan alias/. Untuk menentukan kunci KMS yang berbedaAkun AWS, Anda harus menggunakan kunci ARN atau alias ARN.

Sebelum menjalankan perintah ini, ganti contoh nama alias dengan identifier yang valid untuk kunci KMS. Untuk menjalankan perintah ini, Anda harus memiliki `kms:GetPublicKey` izin pada kunci KMS.

```
$ aws kms get-public-key --key-id alias/example_RSA_3072
```

```
{
    "KeySpec": "RSA_3072",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ],
    "PublicKey": "MIIBojANBgkqhkiG..."
}
```

Mengidentifikasi kunci KMS asimetris

Untuk menentukan apakah kunci KMS tertentu adalah kunci KMS asimetris, cari jenis kunci atau spesifikasi kunci. Anda dapat menggunakan konsol AWS KMS atau API AWS KMS.

Beberapa metode ini juga menunjukkan aspek lain dari konfigurasi kriptografi kunci KMS, termasuk penggunaan kunci dan enkripsi atau algoritma penandatanganan yang didukung oleh kunci KMS. Anda dapat melihat konfigurasi kriptografi kunci KMS yang ada, tetapi Anda tidak dapat mengubahnya.

Untuk informasi umum tentang melihat tombol KMS, termasuk menyortir, memfilter, dan memilih kolom untuk tampilan konsol Anda, lihat. Melihat tombol KMS di konsol

Topik

- Menemukan tipe kunci dalam tabel kunci KMS
- Menemukan jenis kunci pada halaman detail
- Menemukan spesifikasi kunci menggunakan API AWS KMS

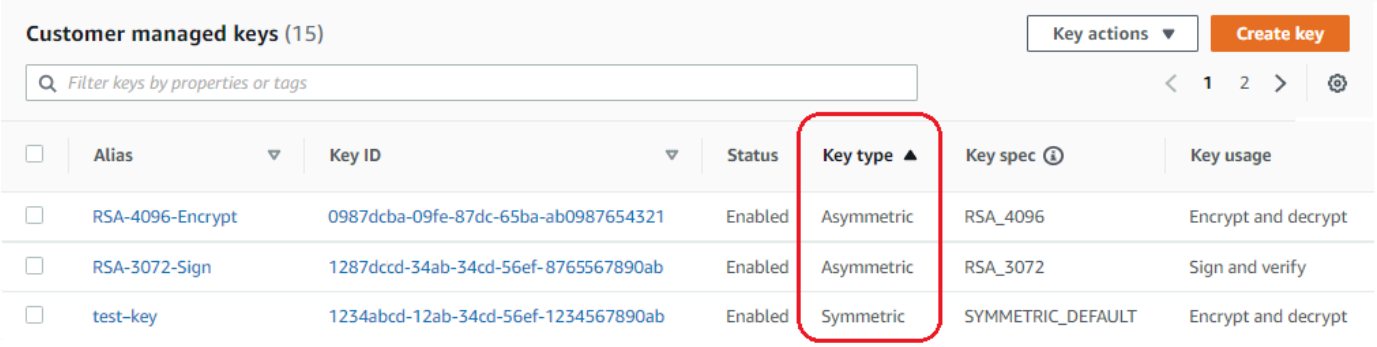
Menemukan tipe kunci dalam tabel kunci KMS

Di AWS KMS konsol, kolom Key type menunjukkan apakah setiap tombol KMS simetris atau asimetris. Anda dapat menambahkan kolom Key type ke tabel kunci KMS pada kunci atau Kunci yang dikelola AWShalaman yang dikelola Pelanggan di konsol.

Untuk mengidentifikasi kunci KMS simetris dan asimetris di tabel kunci KMS Anda, gunakan prosedur berikut.

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih kunci yang dikelola AWS.
4. Kolom Key type menunjukkan apakah setiap tombol KMS simetris atau asimetris. Anda juga dapat mengurutkan dan memfilter berdasarkan nilai Tipe kunci.

   Jika kolom Jenis kunci tidak muncul di tabel kunci KMS Anda, pilih ikon roda gigi di sudut kanan atas halaman, pilih Jenis kunci, lalu pilih Konfirmasi. Anda juga dapat menambahkan kolom Spesifikasi kunci dan Penggunaan kunci.



Menemukan jenis kunci pada halaman detail

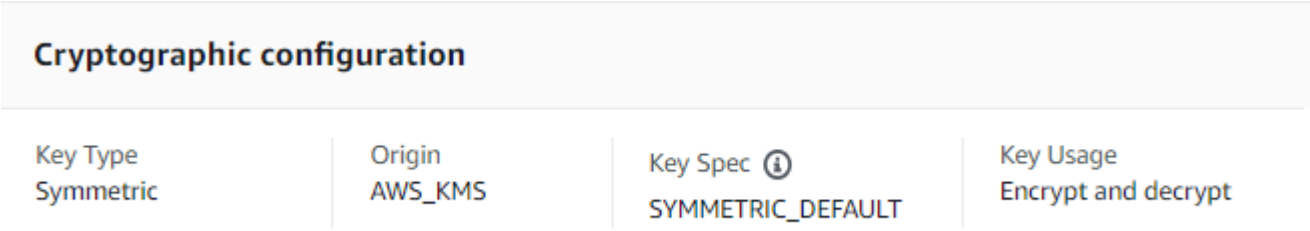
Di AWS KMS konsol, halaman detail untuk setiap kunci KMS mencakup tab Konfigurasi Kriptografi yang menampilkan jenis kunci (simetris atau asimetris) dan detail kriptografi lainnya tentang kunci KMS.

Untuk mengidentifikasi kunci KMS simetris dan asimetris pada halaman detail untuk kunci KMS, gunakan prosedur berikut.

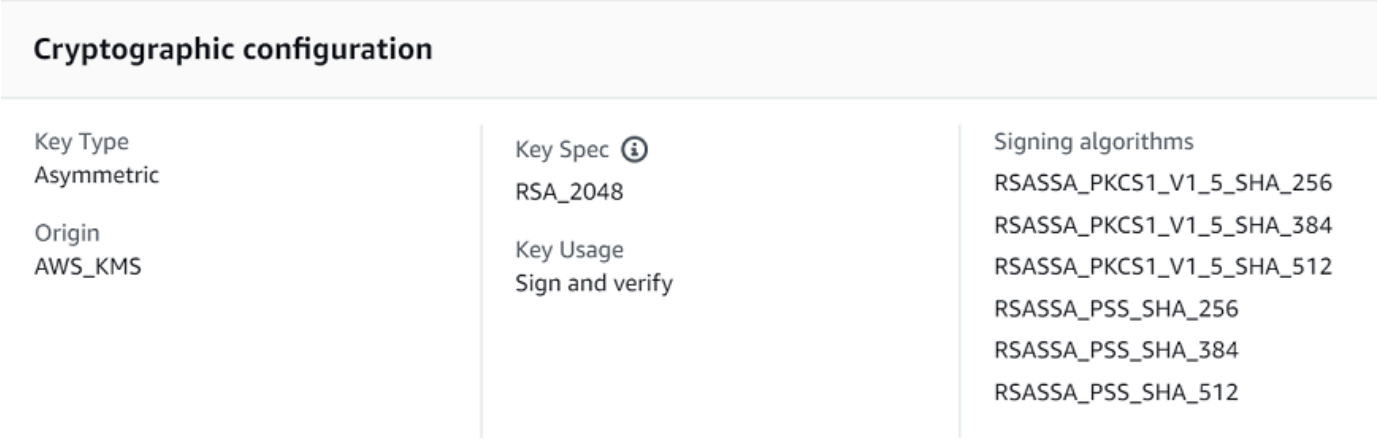
1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Untuk melihat tombol di akun yang Anda buat dan kelola, di panel navigasi pilih Kunci yang dikelola pelanggan. Untuk melihat tombol di akun yang dibuat dan dikelola AWS untuk Anda, di panel navigasi pilih kunci yang dikelola AWS.
4. Pilih alias atau ID kunci dari kunci KMS.
5. Pilih tab Konfigurasi kriptografi. Tab ada di bawah bagian Konfigurasi umum.

Tab Konfigurasi kriptografi menampilkan Tipe Kunci, yang menunjukkan apakah itu simetris atau asimetris. Ini juga menampilkan detail lain tentang kunci KMS, termasuk Penggunaan Kunci, yang memberi tahu apakah kunci KMS dapat digunakan untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi. Untuk kunci KMS asimetris, ini menampilkan algoritma enkripsi atau algoritma penandatanganan yang didukung oleh kunci KMS.

Misalnya, berikut ini adalah contoh tab konfigurasi kriptografi untuk kunci KMS enkripsi simetris.



Berikut ini adalah contoh tab konfigurasi kriptografi untuk kunci KMS RSA asimetris yang digunakan untuk penandatanganan dan verifikasi.



Menemukan spesifikasi kunci menggunakan API AWS KMS

Untuk menentukan apakah kunci KMS simetris atau asimetris, gunakan operasi. DescribeKey `KeySpec`Bidang dalam respons berisi spesifikasi kunci kunci KMS. Untuk kunci KMS enkripsi simetris, nilainya adalah. `KeySpec` `SYMMETRIC_DEFAULT` Nilai lain menunjukkan kunci KMS asimetris atau kunci HMAC KMS.

> **Note**
>
> `CustomerMasterKeySpec`Anggota tidak digunakan lagi. Sebaliknya, gunakan `KeySpec`. Untuk mencegah perubahan yang melanggar, `DescribeKey` respons termasuk `KeySpec` dan `CustomerMasterKeySpec` anggota dengan nilai yang sama.

Misalnya, `DescribeKey` mengembalikan respons berikut untuk kunci KMS enkripsi simetris. Nilai `KeySpec` adalah `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"
    ]
  }
}
```

`DescribeKey`Respons untuk kunci KMS RSA asimetris yang digunakan dalam penandatanganan dan verifikasi terlihat mirip dengan contoh ini. `KeySpec`Nilainya adalah RSA_2048 dan adalah. `KeyUsage` `SIGN_VERIFY` `SigningAlgorithms`Elemen mencantumkan algoritma penandatanganan yang valid untuk kunci KMS.

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

Spesifikasi kunci asimetris

Topik berikut memberikan informasi teknis tentang spesifikasi utama yang AWS KMS mendukung kunci KMS asimetris. Informasi tentang spesifikasi kunci SYMMETRIC_DEFAULT untuk kunci enkripsi simetris disertakan untuk perbandingan.

Topik

- Spesifikasi kunci RSA
- Spesifikasi kunci kurva elips
- Spesifikasi kunci SM2 (hanya Wilayah China)
- Spesifikasi kunci SYMMETRIC_DEFAULT

Spesifikasi kunci RSA

Saat Anda menggunakan spesifikasi kunci RSA, AWS KMS buat kunci KMS asimetris dengan key pair RSA. Kunci privat tidak pernah membiarkan AWS KMS tidak terenkripsi. Anda dapat

menggunakan kunci publik dalam AWS KMS, atau mengunduh kunci publik untuk digunakan di luar AWS KMS.

> **Warning**
>
> Saat Anda mengenkripsi data di luar AWS KMS, pastikan bahwa Anda dapat mendekripsi ciphertext. Jika Anda menggunakan kunci publik dari kunci KMS yang telah dihapusAWS KMS, kunci publik dari kunci KMS yang dikonfigurasi untuk penandatanganan dan verifikasi, atau algoritma enkripsi yang tidak didukung oleh kunci KMS, data tidak dapat dipulihkan.

DiAWS KMS, Anda dapat menggunakan kunci KMS asimetris dengan pasangan kunci RSA untuk enkripsi dan dekripsi, atau penandatanganan dan verifikasi, tetapi tidak keduanya. Properti ini, yang dikenal sebagai penggunaan kunci, ditentukan secara terpisah dari spesifikasi kunci, tetapi Anda harus membuat keputusan tersebut sebelum memilih spesifikasi kunci.

AWS KMS mendukung spesifikasi kunci RSA berikut untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi:

- RSA_2048
- RSA_3072
- RSA_4096

Spesifikasi kunci RSA berbeda dengan panjang kunci RSA dalam bit. Spesifikasi kunci RSA yang Anda pilih mungkin ditentukan oleh standar keamanan atau persyaratan tugas Anda. Secara umum, gunakan kunci terbesar yang praktis dan terjangkau untuk tugas Anda. Operasi kriptografi pada kunci KMS dengan spesifikasi kunci RSA yang berbeda diberi harga berbeda. Untuk informasi tentang harga AWS KMS, lihat Harga Layanan Pengelolaan Kunci AWS. Untuk informasi tentang kuota permintaan, lihat Kuota permintaan.

Spesifikasi kunci RSA untuk enkripsi dan dekripsi

Ketika kunci KMS asimetris RSA digunakan untuk enkripsi dan dekripsi, Anda mengenkripsi dengan kunci publik dan mendekripsi dengan kunci pribadi. Ketika Anda memanggil `Encrypt` operasi AWS KMS untuk kunci RSA KMS, AWS KMS gunakan kunci publik di RSA key pair dan algoritma enkripsi yang Anda tentukan untuk mengenkripsi data Anda. Untuk mendekripsi ciphertext, panggil `Decrypt` operasi dan tentukan kunci KMS dan algoritma enkripsi yang sama. AWS KMSkemudian menggunakan kunci pribadi di RSA key pair untuk mendekripsi data Anda.

Anda juga dapat mengunduh kunci publik dan menggunakannya untuk mengenkripsi data di luar AWS KMS. Pastikan untuk menggunakan algoritma enkripsi yang AWS KMS mendukung kunci RSA KMS. Untuk mendekripsi ciphertext, panggil `Decrypt` fungsi dengan kunci KMS dan algoritma enkripsi yang sama.

AWS KMSmendukung dua algoritma enkripsi untuk kunci KMS dengan spesifikasi kunci RSA. Algoritme ini, yang ditentukan dalam PKCS #1 v2.2, berbeda dalam fungsi hash yang digunakan secara internal. Di AWS KMS, algoritme RSAES_OAEP selalu menggunakan fungsi hash yang sama untuk kedua tujuan hashing dan untuk fungsi pembuatan mask (MGF1). Anda diminta untuk menentukan algoritme enkripsi saat memanggil operasi Enkripsi dan Dekripsi. Anda dapat memilih algoritme yang berbeda untuk setiap permintaan.

Algoritme enkripsi yang didukung untuk spesifikasi kunci RSA

Enkripsi algoritme	Deskripsi algoritme
RSAES_OAEP_SHA_1	PKCS #1 v2.2, Bagian 7.1. Enkripsi RSA dengan OAEP Padding menggunakan SHA-1 untuk hash dan di fungsi pembuatan mask MGF1 beserta label kosong.
RSAES_OAEP_SHA_256	PKCS #1, Bagian 7.1. Enkripsi RSA dengan OAEP Padding menggunakan SHA-256 untuk hash dan di fungsi pembuatan mask MGF1 beserta label kosong.

Anda tidak dapat mengkonfigurasi kunci KMS untuk menggunakan algoritma enkripsi tertentu. Namun, Anda dapat menggunakan kondisi EncryptionAlgorithm kebijakan kms: untuk menentukan algoritme enkripsi yang diizinkan untuk digunakan oleh prinsipal dengan kunci KMS.

Untuk mendapatkan algoritma enkripsi untuk kunci KMS, lihat konfigurasi kriptografi kunci KMS di AWS KMS konsol atau gunakan operasi. DescribeKey AWS KMSjuga menyediakan spesifikasi kunci dan algoritma enkripsi saat Anda mengunduh kunci publik Anda, baik di AWS KMS konsol atau dengan menggunakan operasi. GetPublicKey

Anda dapat memilih spesifikasi kunci RSA berdasarkan panjang data teks biasa yang dapat Anda enkripsi di setiap permintaan. Tabel berikut menunjukkan ukuran maksimum, dalam byte, teks biasa yang dapat Anda enkripsi dalam satu panggilan ke operasi Enkripsi. Nilai tersebut berbeda dengan

spesifikasi kunci dan algoritme enkripsi. Untuk membandingkan, Anda dapat menggunakan kunci KMS enkripsi simetris untuk mengenkripsi hingga 4096 byte sekaligus.

Untuk menghitung panjang teks biasa maksimum dalam byte untuk algoritme ini, gunakan rumus berikut: (*`key_size_in_bits`* / 8) - (2 * *`hash_length_in_bits`*/8) - 2. Misalnya, untuk RSA_2048 dengan SHA-256, ukuran teks biasa maksimum dalam byte adalah (2048/8) - (2 * 256/8) -2 = 190.

Ukuran teks biasa maksimum (dalam byte) dalam operasi Enkripsi

	Algoritme enkripsi	
Spesifikasi kunci	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

Spesifikasi kunci RSA untuk penandatanganan dan verifikasi

Ketika kunci KMS asimetris RSA digunakan untuk penandatanganan dan verifikasi, Anda membuat tanda tangan untuk pesan dengan kunci pribadi dan memverifikasi tanda tangan dengan kunci publik.

Saat Anda memanggil `Sign` operasi AWS KMS untuk kunci KMS asimetris, AWS KMS gunakan kunci pribadi di key pair RSA, pesan, dan algoritma penandatanganan yang Anda tentukan, untuk menghasilkan tanda tangan. Untuk memverifikasi tanda tangan, hubungi operasi Verifikasi. Tentukan tanda tangan, ditambah kunci KMS, pesan, dan algoritma penandatanganan yang sama. AWS KMSkemudian menggunakan kunci publik di RSA key pair untuk memverifikasi tanda tangan. Anda juga dapat mengunduh kunci publik dan menggunakannya untuk memverifikasi tanda tangan di luar AWS KMS.

AWS KMSmendukung algoritma penandatanganan berikut untuk semua kunci KMS dengan spesifikasi kunci RSA. Anda diminta untuk menentukan algoritme penandatanganan saat memanggil operasi Tanda dan Verifikasi. Anda dapat memilih algoritme yang berbeda untuk setiap permintaan. Saat menandatangani dengan pasangan kunci RSA, algoritma RSASSA-PSS lebih disukai. Kami menyertakan algoritma RSASSA-SAPKCS1-v1_5 untuk kompatibilitas dengan aplikasi yang ada.

Algoritme penandatanganan yang didukung untuk spesifikasi kunci RSA

Algoritme penandatanganan	Deskripsi algoritme
RSASSA_PSS_SHA_256	PKCS #1 v2.2, Bagian 8.1, tanda tangan RSA dengan padding PSS menggunakan SHA-256 untuk penyerapan pesan dan fungsi pembuatan mask MGF1 beserta salt 256-bit
RSASSA_PSS_SHA_384	PKCS #1 v2.2, Bagian 8.1, tanda tangan RSA dengan padding PSS menggunakan SHA-384 untuk penyerapan pesan dan fungsi pembuatan mask MGF1 beserta salt 384-bit
RSASSA_PSS_SHA_512	PKCS #1 v2.2, Bagian 8.1, tanda tangan RSA dengan padding PSS menggunakan SHA-512 untuk penyerapan pesan dan fungsi pembuatan mask MGF1 beserta salt 512-bit
RSASSA_PKCS1_V1_5_SHA_256	PKCS #1 v2.2, Bagian 8.2, tanda tangan RSA dengan PKCS #1v1.5 Padding dan SHA-256
RSASSA_PKCS1_V1_5_SHA_384	PKCS #1 v2.2, Bagian 8.2, tanda tangan RSA dengan PKCS #1v1.5 Padding dan SHA-384
RSASSA_PKCS1_V1_5_SHA_512	PKCS #1 v2.2, Bagian 8.2, tanda tangan RSA dengan PKCS #1v1.5 Padding dan SHA-512

Anda tidak dapat mengonfigurasi kunci KMS untuk menggunakan algoritma penandatanganan tertentu. Namun, Anda dapat menggunakan kondisi SigningAlgorithm kebijakan kms: untuk menentukan algoritme penandatanganan yang diizinkan untuk digunakan oleh prinsipal dengan kunci KMS.

Untuk mendapatkan algoritma penandatanganan untuk kunci KMS, lihat konfigurasi kriptografi kunci KMS di AWS KMS konsol atau dengan menggunakan operasi. DescribeKey AWS KMSjuga menyediakan spesifikasi kunci dan algoritma penandatanganan saat Anda mengunduh kunci publik Anda, baik di AWS KMS konsol atau dengan menggunakan operasi. GetPublicKey

Spesifikasi kunci kurva elips

Saat Anda menggunakan spesifikasi kunci eliptic curve (ECC), AWS KMS buat kunci KMS asimetris dengan key pair ECC untuk penandatanganan dan verifikasi. Kunci privat yang membuat tanda tangan selalu mengenkripsi AWS KMS. Anda dapat menggunakan kunci publik untuk memverifikasi tanda tangan dalam AWS KMS, atau mengunduh kunci publik untuk digunakan di luar AWS KMS.

AWS KMSmendukung spesifikasi kunci ECC berikut untuk kunci KMS asimetris.

- Pasangan kunci kurva elips (penandatanganan dan verifikasi) rekomendasi NIST Asimetris
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- Pasangan kunci kurva elips asimetri lain (penandatanganan dan verifikasi)
  - ECC_SECG_P256K1 (secp256k1), biasa digunakan untuk mata uang kripto.

Spesifikasi kunci ECC yang Anda pilih mungkin ditentukan oleh standar keamanan Anda atau persyaratan tugas Anda. Secara umum, gunakan kurva dengan poin terbanyak yang praktis dan terjangkau untuk tugas Anda.

Jika Anda membuat kunci KMS asimetris untuk digunakan dengan cryptocurrency, gunakan spesifikasi kunci ECC_SECG_P256K1. Anda juga dapat menggunakan spesifikasi kunci ini untuk tujuan lain, tetapi spesifikasi ini diperlukan untuk Bitcoin, dan mata uang kripto lainnya.

Kunci KMS dengan spesifikasi kunci ECC yang berbeda diberi harga berbeda dan tunduk pada kuota permintaan yang berbeda. Untuk informasi selengkapnya tentang harga AWS KMS, lihat Harga AWS Key Management Service. Untuk informasi tentang kuota permintaan, lihat Kuota permintaan.

Tabel berikut menunjukkan algoritme penandatanganan yang didukung AWS KMS untuk masing-masing spesifikasi kunci ECC. Anda tidak dapat mengonfigurasi kunci KMS untuk menggunakan algoritma penandatanganan tertentu. Namun, Anda dapat menggunakan kondisi SigningAlgorithm kebijakan kms: untuk menentukan algoritme penandatanganan yang diizinkan untuk digunakan oleh prinsipal dengan kunci KMS.

Algoritme penandatanganan yang didukung untuk spesifikasi kunci ECC

Spesifikasi kunci	Algoritme penandatanganan	Deskripsi algoritme
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Bagian 6.4, tanda tangan ECDSA menggunakan kurva yang ditentukan oleh kunci dan SHA-256 untuk penyerapan pesan.
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, Bagian 6.4, tanda tangan ECDSA menggunakan kurva yang ditentukan oleh kunci dan SHA-384 untuk penyerapan pesan.
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, Bagian 6.4, tanda tangan ECDSA menggunakan kurva yang ditentukan oleh kunci dan SHA-512 untuk penyerapan pesan.
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, Bagian 6.4, tanda tangan ECDSA menggunakan kurva yang ditentukan oleh kunci dan SHA-256 untuk penyerapan pesan.

Spesifikasi kunci SM2 (hanya Wilayah China)

Spesifikasi kunci SM2 adalah spesifikasi kunci kurva elips yang ditentukan dalam rangkaian spesifikasi GM/T yang diterbitkan oleh Kantor Administrasi Kriptografi Komersil Negara (OSCCA) China. Spesifikasi kunci SM2 hanya tersedia di Wilayah China. Saat Anda menggunakan spesifikasi

kunci SM2, AWS KMS buat kunci KMS asimetris dengan key pair SM2. Anda dapat menggunakan SM2 key pair di dalamAWS KMS, atau mengunduh kunci publik untuk digunakan di luar. AWS KMS

Berbeda dengan spesifikasi kunci ECC, Anda dapat menggunakan kunci SM2 KMS untuk penandatanganan dan verifikasi, atau enkripsi dan dekripsi. Anda harus menentukan penggunaan kunci saat Anda membuat kunci KMS, dan Anda tidak dapat mengubahnya setelah kunci dibuat.

AWS KMSmendukung enkripsi SM2 dan algoritma penandatanganan berikut:

- Algoritma enkripsi SM2PKE

  SM2PKE adalah algoritma enkripsi berbasis kurva elips yang didefinisikan oleh OSCCA dalam GM/T 0003.4-2012.

- Algoritma penandatanganan SM2DSA

  SM2DSA adalah algoritma penandatanganan berbasis kurva elips yang didefinisikan oleh OSCCA dalam GM/T 0003.2-2012. SM2DSA memerlukan ID pembeda yang di-hash dengan algoritma hashing SM3 dan kemudian dikombinasikan dengan pesan, atau intisari pesan, yang Anda kirimkan. AWS KMS Nilai gabungan ini kemudian di-hash dan ditandatangani oleh. AWS KMS

Operasi offline dengan SM2 (hanya Wilayah China)

Anda dapat mengunduh kunci publik dari SM2 key pair Anda untuk digunakan dalam operasi offline, yaitu operasi di luar. AWS KMS Namun, saat menggunakan kunci publik SM2 offline, Anda mungkin perlu melakukan konversi dan perhitungan tambahan secara manual. Operasi SM2DSA mungkin mengharuskan Anda untuk memberikan ID yang membedakan atau menghitung intisari pesan. Operasi enkripsi SM2PKE mungkin mengharuskan Anda untuk mengonversi output ciphertext mentah ke format yang dapat diterima. AWS KMS

Untuk membantu Anda dengan operasi ini, `SM2OfflineOperationHelper` kelas untuk Java memiliki metode yang melakukan tugas untuk Anda. Anda dapat menggunakan kelas pembantu ini sebagai model untuk penyedia kriptografi lainnya.

> **Important**
>
> Kode `SM2OfflineOperationHelper` referensi dirancang agar kompatibel dengan Bouncy Castle versi 1.68. Untuk bantuan dengan versi lain, hubungi bouncycastle.org.

Verifikasi offline dengan pasangan kunci SM2 (hanya Wilayah China)

Untuk memverifikasi tanda tangan di luar AWS KMS dengan kunci publik SM2, Anda harus menentukan ID pembeda. Saat Anda meneruskan pesan mentah, ke Sign API `MessageType:RAW`, AWS KMS menggunakan ID pembeda default, yang ditentukan oleh OSCCA di GM/T 0009-2012. `1234567812345678` Anda tidak dapat menentukan ID pembeda Anda sendiri di dalamnyaAWS KMS.

Namun, jika Anda membuat intisari pesan di luarAWS, Anda dapat menentukan ID pembeda Anda sendiri, lalu meneruskan intisari pesan `MessageType:DIGEST`, untuk AWS KMS menandatangani. Untuk melakukan ini, ubah `DEFAULT_DISTINGUISHING_ID` nilai di `SM2OfflineOperationHelper` kelas. ID pembeda yang Anda tentukan dapat berupa string apa pun hingga 8,192 karakter. Setelah AWS KMS menandatangani intisari pesan, Anda memerlukan intisari pesan atau pesan dan ID pembeda yang digunakan untuk menghitung intisari untuk memverifikasinya secara offline.

`SM2OfflineOperationHelper` kelas

Di dalamnyaAWS KMS, konversi ciphertext mentah dan perhitungan intisari pesan SM2DSA terjadi secara otomatis. Tidak semua penyedia kriptografi menerapkan SM2 dengan cara yang sama. Beberapa pustaka, seperti OpenSSL versi 1.1.1 dan yang lebih baru, melakukan tindakan ini secara otomatis. AWS KMSmengkonfirmasi perilaku ini dalam pengujian dengan OpenSSL versi 3.0. Gunakan `SM2OfflineOperationHelper` kelas berikut dengan pustaka, seperti Bouncy Castle, yang mengharuskan Anda melakukan konversi dan perhitungan ini secara manual.

`SM2OfflineOperationHelper`Kelas menyediakan metode untuk operasi offline berikut:

- Perhitungan intisari pesan

  Untuk menghasilkan intisari pesan secara offline yang dapat Anda gunakan untuk verifikasi offline, atau yang dapat diteruskan AWS KMS ke tanda tangan, gunakan `calculateSM2Digest` metode ini. `calculateSM2Digest`Metode ini menghasilkan intisari pesan dengan algoritma hashing SM3. GetPublicKeyAPI mengembalikan kunci publik Anda dalam format biner. Anda harus mengurai kunci biner menjadi Java PublicKey. Berikan kunci publik yang diurai dengan pesan. Metode ini secara otomatis menggabungkan pesan Anda dengan ID pembeda default`1234567812345678`, tetapi Anda dapat mengatur ID pembeda Anda sendiri dengan mengubah nilainya. `DEFAULT_DISTINGUISHING_ID`

- Verifikasi

  Untuk memverifikasi tanda tangan secara offline, gunakan `offlineSM2DSAVerify` metode ini. `offlineSM2DSAVerify`Metode ini menggunakan intisari pesan yang dihitung dari ID pembeda yang ditentukan, dan pesan asli yang Anda berikan untuk memverifikasi tanda tangan digital. GetPublicKeyAPI mengembalikan kunci publik Anda dalam format biner. Anda harus mengurai kunci biner menjadi Java PublicKey. Berikan kunci publik yang diuraikan dengan pesan asli dan tanda tangan yang ingin Anda verifikasi. Untuk detail selengkapnya, lihat Verifikasi offline dengan pasangan kunci SM2.

- Enkripsi

  Untuk mengenkripsi plaintext offline, gunakan metode ini. `offlineSM2PKEEncrypt` Metode ini memastikan ciphertext dalam format AWS KMS dapat mendekripsi. `offlineSM2PKEEncrypt`Metode mengenkripsi plaintext, dan kemudian mengonversi ciphertext mentah yang dihasilkan oleh SM2PKE ke format ASN.1. GetPublicKeyAPI mengembalikan kunci publik Anda dalam format biner. Anda harus mengurai kunci biner menjadi Java PublicKey. Berikan kunci publik yang diurai dengan plaintext yang ingin Anda enkripsi.

  Jika Anda tidak yakin apakah Anda perlu melakukan konversi, gunakan operasi OpenSSL berikut untuk menguji format ciphertext Anda. Jika operasi gagal, Anda perlu mengonversi ciphertext ke format ASN.1.

  ```
  openssl asn1parse -inform DER -in ciphertext.der
  ```

Secara default, `SM2OfflineOperationHelper` kelas menggunakan ID pembeda default`1234567812345678`, saat menghasilkan intisari pesan untuk operasi SM2DSA.

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
```

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
 distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
 "1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
 GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
 message) throws
            NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;

        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
 0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
```

```
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(entla.length +
DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
                        xa.length +
ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
                        .array());

        // Combine hashed distinguishing ID with original message to generate final
digest
        return MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(za.length +
message.length).put(za).put(message)
                        .array());
    }

    // ***offlineSM2DSAVerify***
    // Verify digital signature with SM2 public key
    public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
            final byte [] signature) throws InvalidKeyException {
        final SM2Signer signer = new SM2Signer();
        CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
        cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
        signer.init(false, cipherParameters);
        signer.update(message, 0, message.length);
        return signer.verifySignature(signature);
    }

    // ***offlineSM2PKEEncrypt***
    // Encrypt data with SM2 public key
    public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
plaintext) throws
            NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException, IOException {
        final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
        sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);
```

```
        // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
        final byte [] cipherText = sm2Cipher.doFinal(plaintext);

        // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
        final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
        final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
 7) / 8 * 2 + 1;
        final int sm3HashLength = 32;
        final int xCoordinateInCipherText = 33;
        final int yCoordinateInCipherText = 65;
        byte[] coords = new byte[coordinateLength];
        byte[] sm3Hash = new byte[sm3HashLength];
        byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
 sm3HashLength];

        // Split components out of the ciphertext
        System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
        System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
 sm3HashLength);
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
 0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

Spesifikasi kunci SYMMETRIC_DEFAULT

Spesifikasi kunci default, SYMMETRIC_DEFAULT, adalah spesifikasi kunci untuk kunci KMS enkripsi simetris. Ketika Anda memilih jenis kunci simetris dan penggunaan kunci Enkripsi dan dekripsi di AWS KMS konsol, ia memilih spesifikasi kunci. `SYMMETRIC_DEFAULT` Dalam CreateKeyoperasi, jika Anda tidak menentukan `KeySpec` nilai, SYMMETRIC_DEFAULT dipilih. Jika Anda tidak memiliki

alasan untuk menggunakan spesifikasi kunci yang berbeda, SYMMETRIC_DEFAULT adalah pilihan yang tepat.

SYMMETRIC_DEFAULT saat ini mewakili AES-256-GCM, algoritma simetris berdasarkan Advanced Encryption Standard (AES) dalam Galois Counter Mode (GCM) dengan kunci 256-bit, standar industri untuk enkripsi aman. Ciphertext yang dihasilkan oleh algoritme ini mendukung data terautentikasi tamabahan (AAD), seperti konteks enkripsi, dan GCM menyediakan pemeriksaan integritas tambahan pada ciphertext. Untuk detail teknis, lihat Detail Kriptografi AWS Key Management Service.

Data yang dienkripsi berdasarkan AES-256-GCM dilindungi sekarang dan di masa mendatang. Kriptografer menganggap algoritme ini sebagai tahan kuantum. Komputasi kuantum skala besar yang secara teoritis disiapkan untuk menghadapi masa depan akan menyerang ciphertext yang dibuat berdasarkan kunci 256-bit AES-GCM, sehingga mengurangi keamanan efektif untuk kunci menjadi 128 bit. Namun, tingkat keamanan ini cukup untuk menangkal serangan brute force pada ciphertext AWS KMS.

Satu-satunya pengecualian di Wilayah China, di mana SYMMETRIC_DEFAULT mewakili kunci simetris 128-bit yang menggunakan enkripsi SM4. Anda hanya dapat membuat kunci SM4 128-bit di Wilayah China. Anda tidak dapat membuat kunci KMS AES-GCM 256-bit di Wilayah China.

Anda dapat menggunakan kunci KMS enkripsi simetris AWS KMS untuk mengenkripsi, mendekripsi, dan mengenkripsi ulang data, dan untuk melindungi kunci data dan pasangan kunci data yang dihasilkan. AWSlayanan yang terintegrasi dengan AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda saat istirahat. Anda dapat mengimpor materi kunci Anda sendiri ke kunci KMS enkripsi simetris dan membuat kunci KMS enkripsi simetris di toko kunci khusus. Untuk tabel yang membandingkan operasi yang dapat Anda lakukan pada tombol KMS simetris dan asimetris, lihat Membandingkan kunci KMS Simetris dan Asimetris.

Untuk detail teknis tentang AWS KMS dan kunci enkripsi simetris, lihat Detail AWS Key Management ServiceKriptografi.

Kunci HMAC di AWS KMS

Kunci KMS Kode Otentikasi Pesan Berbasis Hash (HMAC) adalah kunci simetris yang Anda gunakan untuk membuat dan memverifikasi HMAC di dalamnya. AWS KMS Materi kunci unik yang terkait dengan setiap kunci HMAC KMS menyediakan kunci rahasia yang dibutuhkan algoritma HMAC. Anda dapat menggunakan kunci HMAC KMS dengan `GenerateMac` dan `VerifyMac`operasi untuk memverifikasi integritas dan keaslian data di dalamnya. AWS KMS

Algoritma HMAC menggabungkan fungsi hash kriptografi dan kunci rahasia bersama. Mereka mengambil pesan dan kunci rahasia, seperti materi kunci dalam kunci HMAC KMS, dan mengembalikan kode atau tag ukuran tetap yang unik. Jika bahkan satu karakter pesan berubah, atau jika kunci rahasia tidak identik, tag yang dihasilkan sama sekali berbeda. Dengan membutuhkan kunci rahasia, HMAC juga memberikan keaslian; tidak mungkin untuk menghasilkan tag HMAC identik tanpa kunci rahasia. HMAC kadang-kadang disebut tanda tangan simetris, karena mereka bekerja seperti tanda tangan digital, tetapi menggunakan satu kunci untuk penandatanganan dan verifikasi.

Kunci HMAC KMS dan algoritma HMAC yang AWS KMS menggunakan sesuai dengan standar industri yang didefinisikan dalam RFC 2104. AWS KMSGenerateMacOperasi menghasilkan tag HMAC standar. Kunci KMS HMAC dihasilkan dalam modul keamanan AWS KMS perangkat keras yang disertifikasi di bawah Program Validasi Modul Kriptografi FIPS 140-2 (kecuali di Wilayah China (Beijing) dan China (Ningxia)) dan tidak pernah dibiarkan tanpa terenkripsi. AWS KMS Untuk menggunakan kunci HMAC KMS, Anda harus menelepon. AWS KMS

Anda dapat menggunakan kunci HMAC KMS untuk menentukan keaslian pesan, seperti JSON Web Token (JWT), informasi kartu kredit token, atau kata sandi yang dikirimkan. Mereka juga dapat digunakan sebagai Secure Key Derivation Functions (KDFs), terutama dalam aplikasi yang membutuhkan kunci deterministik.

Kunci HMAC KMS memberikan keuntungan dibandingkan HMAC dari perangkat lunak aplikasi karena materi utama dihasilkan dan digunakan sepenuhnya di dalamAWS KMS, tunduk pada kontrol akses yang Anda tetapkan pada kunci.

> **Tip**
>
> Praktik terbaik merekomendasikan agar Anda membatasi waktu selama mekanisme penandatanganan apa pun, termasuk HMAC, efektif. Ini mencegah serangan di mana aktor menggunakan pesan yang ditandatangani untuk menetapkan validitas berulang kali atau lama setelah pesan digantikan. Tag HMAC tidak menyertakan stempel waktu, tetapi Anda dapat menyertakan stempel waktu dalam token atau pesan untuk membantu Anda mendeteksi kapan waktunya untuk menyegarkan HMAC.

Pengguna yang berwenang dapat membuat, mengelola, dan menggunakan kunci HMAC KMS di akun AndaAWS. Ini termasuk mengaktifkan dan menonaktifkan kunci, mengatur dan mengubah alias dan tag, dan menjadwalkan penghapusan kunci HMAC KMS. Anda juga dapat mengontrol akses ke

kunci HMAC KMS menggunakan kebijakan utama, kebijakanIAM, dan hibah. Anda dapat mengaudit semua operasi yang menggunakan atau mengelola kunci KMS HMAC Anda AWS di AWS CloudTrail dalam log. Anda dapat membuat kunci HMAC KMS dengan bahan kunci yang diimpor. Anda juga dapat membuat kunci KMS Multi-wilayah HMAC yang berperilaku seperti salinan kunci KMS HMAC yang sama dalam beberapa. Wilayah AWS

Kunci HMAC KMS hanya mendukung operasi `GenerateMac`dan `VerifyMac`kriptografi. Anda tidak dapat menggunakan kunci HMAC KMS untuk mengenkripsi data atau menandatangani pesan, atau menggunakan jenis kunci KMS lainnya dalam operasi HMAC. Saat Anda menggunakan `GenerateMac` operasi, Anda menyediakan pesan hingga 4.096 byte, kunci HMAC KMS, dan algoritma MAC yang kompatibel dengan spesifikasi kunci HMAC, dan menghitung tag HMAC. `GenerateMac` Untuk memverifikasi tag HMAC, Anda harus menyediakan tag HMAC, dan pesan yang sama, kunci HMAC KMS, dan algoritma MAC yang `GenerateMac` digunakan untuk menghitung tag HMAC asli. `VerifyMac`Operasi menghitung tag HMAC dan memverifikasi bahwa itu identik dengan tag HMAC yang disediakan. Jika input dan tag HMAC yang dihitung tidak identik, verifikasi gagal.

Kunci HMAC KMS tidak mendukung rotasi kunci otomatis dan Anda tidak dapat membuat kunci HMAC KMS di toko kunci khusus.

Jika Anda membuat kunci KMS untuk mengenkripsi data dalam suatu AWS layanan, gunakan kunci enkripsi simetris. Anda tidak dapat menggunakan kunci HMAC KMS.

Daerah

Kunci HMAC KMS didukung di semua Wilayah AWS yang AWS KMS mendukung.

Pelajari selengkapnya

- Untuk bantuan dalam memilih jenis kunci KMS, lihatMemilih tipe kunci KMS.
- Untuk tabel yang membandingkan operasi AWS KMS API yang didukung oleh setiap jenis kunci KMS, lihat. Referensi tipe kunci
- Untuk informasi tentang membuat kunci KMS HMAC Multi-wilayah, lihat. Kunci Multi-Region di AWS KMS
- Untuk memeriksa perbedaan dalam kebijakan kunci default yang ditetapkan AWS KMS konsol untuk kunci HMAC KMS, lihat. the section called “Memungkinkan pengguna kunci untuk menggunakan kunci KMS dengan layanan AWS”
- Untuk informasi tentang harga kunci HMAC KMS, lihat AWS Key Management Service harga.

- Untuk informasi tentang kuota yang berlaku untuk kunci HMAC KMS, lihat dan. Kuota sumber daya Kuota permintaan
- Untuk informasi tentang menghapus kunci HMAC KMS, lihat. Menghapus AWS KMS keys
- Untuk mempelajari cara menggunakan HMAC untuk membuat token web JSON, lihat Cara melindungi HMAC di dalam AWS KMS Blog Keamanan. AWS
- Dengarkan podcast: Memperkenalkan HMAC untuk Podcast AWS Key Management Service Resmi. AWS

Topik

Spesifikasi utama untuk kunci HMAC KMS

AWS KMSmendukung kunci HMAC simetris dalam berbagai panjang. Spesifikasi kunci yang Anda pilih dapat bergantung pada persyaratan keamanan, peraturan, atau bisnis Anda. Panjang kunci menentukan algoritma MAC yang digunakan dalam GenerateMacdan VerifyMacoperasi. Secara umum, kunci yang lebih panjang lebih aman. Gunakan kunci terpanjang yang praktis untuk kasus penggunaan Anda.

Spesifikasi kunci HMAC	Algoritma MAC
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Membuat kunci HMAC KMS

Anda dapat membuat kunci HMAC KMS di AWS KMS konsol, dengan menggunakan CreateKeyAPI, atau dengan menggunakan template. AWS CloudFormation

AWS KMSmendukung beberapa spesifikasi kunci untuk kunci HMAC KMS. Spesifikasi kunci yang Anda pilih mungkin ditentukan oleh peraturan, keamanan, atau persyaratan bisnis. Secara umum, kunci yang lebih panjang lebih tahan terhadap serangan brute-force.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Jika Anda membuat kunci KMS untuk mengenkripsi data dalam suatu AWS layanan, gunakan kunci KMS enkripsi simetris. AWSlayanan yang terintegrasi dengan AWS KMS tidak mendukung kunci KMS asimetris atau kunci KMS HMAC. Untuk bantuan dalam membuat kunci KMS enkripsi simetris, lihat. Membuat kunci

Pelajari selengkapnya

- Untuk menentukan jenis kunci KMS yang akan dibuat, lihatMemilih tipe kunci KMS.
- Anda dapat menggunakan prosedur yang dijelaskan dalam topik ini untuk membuat kunci KMS HMAC utama Multi-wilayah. Untuk mereplikasi kunci HMAC Multi-wilayah, lihat. the section called “Membuat kunci replika”
- Untuk informasi tentang izin yang diperlukan untuk membuat kunci KMS, lihat. Izin untuk membuat kunci KMS
- Untuk informasi tentang menggunakan AWS CloudFormation template untuk membuat kunci HMAC KMS, lihat AWS::KMS::Keydi AWS CloudFormationPanduan Pengguna.

Topik

- Membuat kunci HMAC KMS (konsol)
- Membuat kunci HMAC KMS (API) AWS KMS

Membuat kunci HMAC KMS (konsol)

Anda dapat menggunakan tombol AWS Management Console untuk membuat kunci HMAC KMS. Kunci HMAC KMS adalah kunci simetris dengan penggunaan kunci Hasilkan dan verifikasi MAC. Anda juga dapat membuat kunci HMAC Multi-wilayah.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Untuk Tipe Kunci, pilih Simetris.

   Kunci HMAC KMS simetris. Anda menggunakan kunci yang sama untuk menghasilkan dan memverifikasi tag HMAC.
6. Untuk penggunaan Kunci, pilih Hasilkan dan verifikasi MAC.

   Menghasilkan dan memverifikasi MAC adalah satu-satunya penggunaan kunci yang valid untuk kunci HMAC KMS.

   > **Note**
   >
   > Penggunaan kunci ditampilkan untuk kunci simetris hanya jika kunci HMAC KMS didukung di Wilayah yang Anda pilih.
7. Pilih spesifikasi (Spesifikasi kunci) untuk kunci HMAC KMS Anda.

   Spesifikasi kunci yang Anda pilih dapat ditentukan oleh peraturan, keamanan, atau persyaratan bisnis. Secara umum, kunci yang lebih panjang lebih aman.
8. Untuk membuat kunci HMAC utama Multi-wilayah, di Opsi lanjutan, pilih kunci Multi-Region. Properti bersama yang Anda tentukan untuk kunci KMS ini, seperti jenis kunci dan penggunaan kunci, akan dibagikan dengan kunci replika. Untuk detailnya, lihat Membuat kunci multi-Wilayah.

   Anda tidak dapat menggunakan prosedur ini untuk membuat kunci replika. Untuk membuat kunci HMAC replika Multi-wilayah, ikuti petunjuk untuk membuat kunci replika.
9. Pilih Berikutnya.

10. Masukkan alias untuk tombol KMS. Nama alias tidak dapat dimulai dengan **`aws/`**. **`aws/`**Awalan dicadangkan oleh Amazon Web Services untuk mewakili Kunci yang dikelola AWS di akun Anda.

    Kami menyarankan Anda menggunakan alias yang mengidentifikasi kunci KMS sebagai kunci HMAC, seperti. `HMAC/test-key` Ini akan memudahkan Anda untuk mengidentifikasi kunci HMAC Anda di AWS KMS konsol tempat Anda dapat mengurutkan dan memfilter kunci berdasarkan tag dan alias, tetapi tidak berdasarkan spesifikasi kunci atau penggunaan kunci.

    Alias diperlukan saat Anda membuat kunci KMS di file. AWS Management Console Anda tidak dapat menentukan alias ketika Anda menggunakan CreateKeyoperasi, tetapi Anda dapat menggunakan konsol atau CreateAliasoperasi untuk membuat alias untuk kunci KMS yang ada. Untuk detailnya, lihat Menggunakan alias.

11. (Opsional) Masukkan deskripsi untuk tombol KMS.

    Masukkan deskripsi yang menjelaskan jenis data yang Anda rencanakan untuk dilindungi atau aplikasi yang Anda rencanakan untuk digunakan dengan kunci KMS.

    Anda dapat menambahkan deskripsi sekarang atau memperbaruinya kapan saja kecuali status kuncinya adalah `Pending Deletion` atau`Pending Replica Deletion`. Untuk menambah, mengubah, atau menghapus deskripsi kunci terkelola pelanggan yang ada, edit deskripsi di AWS Management Console atau gunakan UpdateKeyDescriptionoperasi.

12. (Opsional) Masukkan kunci tag dan nilai tag opsional. Untuk menambahkan lebih dari satu tag ke tombol KMS, pilih Tambah tag.

    Pertimbangkan untuk menambahkan tag yang mengidentifikasi kunci sebagai kunci HMAC, seperti. `Type=HMAC` Ini akan memudahkan Anda untuk mengidentifikasi kunci HMAC Anda di AWS KMS konsol tempat Anda dapat mengurutkan dan memfilter kunci berdasarkan tag dan alias, tetapi tidak berdasarkan spesifikasi kunci atau penggunaan kunci.

    Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS

13. Pilih Berikutnya.
14. Pilih pengguna IAM dan peran yang dapat mengelola kunci KMS.

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini. Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk memberikan izin kepada prinsipal lain untuk mengelola kunci KMS. Untuk detailnya, lihat the section called “Kebijakan kunci default”.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

15. (Opsional) Untuk mencegah pengguna dan peran IAM yang dipilih menghapus kunci KMS ini, di bagian Penghapusan kunci di bagian bawah halaman, kosongkan kotak centang Izinkan administrator kunci untuk menghapus kunci ini.
16. Pilih Berikutnya.
17. Pilih pengguna IAM dan peran yang dapat menggunakan kunci KMS untuk operasi kriptografi.

> **Note**
>
> Kebijakan kunci ini memberikan kontrol Akun AWS penuh atas kunci KMS ini. Ini memungkinkan administrator akun untuk menggunakan kebijakan IAM untuk memberikan izin kepada prinsipal lain untuk menggunakan kunci KMS dalam operasi kriptografi. Untuk detailnya, lihat the section called “Kebijakan kunci default”.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

18. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, dalam bagian Lainnya Akun AWS di bawah halaman, pilih Tambahkan Akun AWS lain dan masukkan nomor identifikasi Akun AWS akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Untuk mengizinkan prinsipal di akun eksternal menggunakan kunci KMS, Administrator akun eksternal harus membuat kebijakan IAM yang memberikan izin ini. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

19. Pilih Selanjutnya.
20. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
21. Pilih Selesai untuk membuat kunci HMAC KMS.

Membuat kunci HMAC KMS (API) AWS KMS

Anda dapat menggunakan CreateKeyoperasi untuk membuat kunci HMAC KMS. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Saat Anda membuat kunci HMAC KMS, Anda harus menentukan `KeySpec` parameter, yang menentukan jenis kunci KMS. Selain itu, Anda harus menentukan `KeyUsage` nilai GENERATE_VERIFY_MAC, meskipun itu satu-satunya nilai penggunaan kunci yang valid untuk kunci HMAC. Untuk membuat kunci Multi-region HMAC KMS, tambahkan `MultiRegion` parameter dengan nilai. `true` Anda tidak dapat mengubah properti ini setelah kunci KMS dibuat.

`CreateKey`Operasi tidak memungkinkan Anda menentukan alias, tetapi Anda dapat menggunakan CreateAliasoperasi untuk membuat alias untuk kunci KMS baru Anda. Kami menyarankan Anda menggunakan alias yang mengidentifikasi kunci KMS sebagai kunci HMAC, seperti. `HMAC/test-key` Ini akan memudahkan Anda untuk mengidentifikasi kunci HMAC Anda di AWS KMS konsol tempat Anda dapat mengurutkan dan memfilter kunci berdasarkan alias, tetapi tidak berdasarkan spesifikasi kunci atau penggunaan kunci.

Jika Anda mencoba untuk membuat kunci HMAC KMS Wilayah AWS di mana kunci HMAC tidak didukung, operasi mengembalikan `CreateKey` `UnsupportedOperationException`

Contoh berikut menggunakan `CreateKey` operasi untuk membuat kunci KMS HMAC 512-bit.

```
$ aws kms create-key --key-spec HMAC_512 --key-usage GENERATE_VERIFY_MAC
```

```
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1669973196.214,
        "MultiRegion": false,
        "KeySpec": "HMAC_512",
        "CustomerMasterKeySpec": "HMAC_512",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_512"
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

Mengontrol akses ke kunci HMAC KMS

Untuk mengontrol akses ke kunci HMAC KMS, Anda menggunakan kebijakan kunci, yang diperlukan untuk setiap kunci KMS. Anda juga dapat menggunakan kebijakan dan hibah IAM.

Kebijakan kunci default untuk kunci HMAC yang dibuat di AWS KMS konsol memberikan izin kepada pengguna kunci untuk memanggil `GenerateMac`dan `VerifyMac`operasi. Namun, itu tidak termasuk pernyataan kebijakan utama yang dirancang untuk menggunakan hibah dengan AWS layanan. Jika Anda membuat kunci HMAC dengan menggunakan `CreateKey`operasi, Anda harus menentukan izin ini dalam kebijakan kunci atau kebijakan IAM.

Anda dapat menggunakan kunci kondisi AWS global dan tombol AWS KMS kondisi untuk memperbaiki dan membatasi izin ke kunci HMAC. Misalnya, Anda dapat menggunakan tombol `kms:ResourceAliases`kondisi untuk mengontrol akses ke AWS KMS operasi berdasarkan alias yang terkait dengan kunci HMAC. Ketentuan AWS KMS kebijakan berikut berguna untuk kebijakan pada kunci HMAC.

- Gunakan tombol `kms:MacAlgorithm`kondisi untuk membatasi algoritme yang dapat diminta oleh prinsipal saat mereka memanggil dan operasi. `GenerateMacVerifyMac` Misalnya, Anda dapat

mengizinkan prinsipal untuk memanggil `GenerateMac` operasi tetapi hanya ketika algoritma MAC dalam permintaan tersebut. `HMAC_SHA_384`

- Gunakan tombol `kms:KeySpec`kondisi untuk mengizinkan atau mencegah prinsipal membuat jenis kunci HMAC tertentu. Misalnya, untuk mengizinkan prinsipal membuat hanya kunci HMAC, Anda dapat mengizinkan CreateKeyoperasi, tetapi gunakan `kms:KeySpec` kondisi untuk mengizinkan hanya kunci dengan spesifikasi kunci. `HMAC_384`

  Anda juga dapat menggunakan tombol `kms:KeySpec` kondisi untuk mengontrol akses ke operasi lain pada kunci KMS berdasarkan spesifikasi kunci kunci. Misalnya, Anda dapat mengizinkan prinsipal untuk menjadwalkan dan membatalkan penghapusan kunci hanya pada kunci KMS dengan spesifikasi kunci. `HMAC_256`

- Gunakan tombol `kms:KeyUsage`kondisi untuk mengizinkan atau mencegah prinsipal membuat kunci HMAC apa pun. Misalnya, untuk mengizinkan prinsipal membuat hanya kunci HMAC, Anda dapat mengizinkan CreateKeyoperasi, tetapi gunakan `kms:KeyUsage` kondisi untuk mengizinkan hanya kunci dengan penggunaan kunci. `GENERATE_VERIFY_MAC`

  Anda juga dapat menggunakan tombol `kms:KeyUsage` kondisi untuk mengontrol akses ke operasi lain pada kunci KMS berdasarkan penggunaan kunci kunci. Misalnya, Anda dapat mengizinkan prinsipal untuk mengaktifkan dan menonaktifkan hanya pada kunci KMS dengan penggunaan kunci. `GENERATE_VERIFY_MAC`

Anda juga dapat membuat hibah untuk `GenerateMac`dan `VerifyMac`operasi, yang merupakan operasi hibah. Namun, Anda tidak dapat menggunakan batasan hibah konteks enkripsi dalam hibah untuk kunci HMAC. Format tag HMAC tidak mendukung nilai konteks enkripsi.

Melihat Kunci HMAC KMS

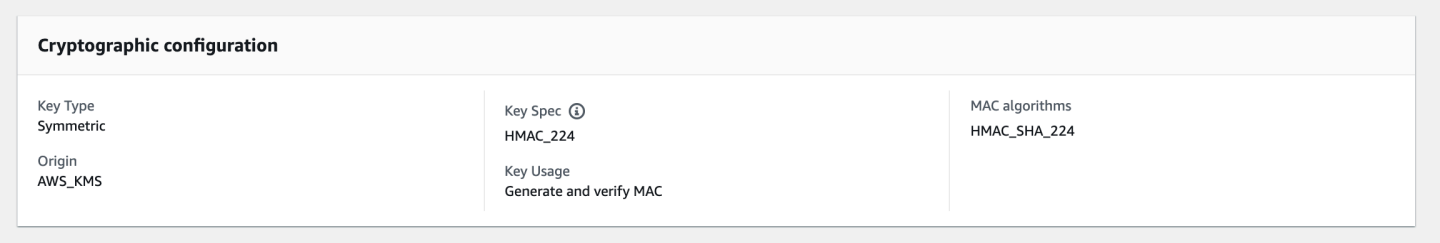
Anda dapat melihat kunci HMAC KMS di AWS KMS konsol atau dengan menggunakan API. DescribeKey Anda dapat memantau penggunaan kunci KMS HMAC Anda di AWS CloudTraillog dan di Amazon. CloudWatch Untuk petunjuk dasar tentang melihat tombol KMS, lihatMelihat kunci.

Anda dapat membedakan kunci HMAC KMS dari jenis kunci KMS lainnya dengan spesifikasi kunci mereka, yang dimulai dengan, atau penggunaan kunci merekaHMAC, yang selalu Menghasilkan dan memverifikasi MAC (). `GENERATE_VERIFY_MAC`

Kunci KMS HMAC disertakan dalam tabel pada halaman kunci yang dikelola Pelanggan di konsol. AWS KMS Namun, Anda tidak dapat mengurutkan atau memfilter kunci KMS berdasarkan spesifikasi

kunci atau penggunaan kunci. Untuk membuatnya lebih mudah untuk menemukan kunci HMAC Anda, tetapkan mereka alias atau tag yang berbeda. Kemudian Anda dapat mengurutkan atau memfilter berdasarkan alias atau tag.

Pada halaman detail kunci untuk kunci HMAC KMS, Anda dapat menemukan detail konfigurasinya di tab konfigurasi Kriptografi.



Kunci Multi-Region di AWS KMS

AWS KMS mendukung kunci Multi-region, yang berbeda Wilayah AWS yang dapat digunakan AWS KMS keys secara bergantian — seolah-olah Anda memiliki kunci yang sama di beberapa Wilayah. Setiap set kunci Multi-wilayah terkait memiliki materi kunci dan ID kunci yang sama, sehingga Anda dapat mengenkripsi data dalam satu Wilayah AWS dan mendekripsi dengan cara yang berbeda Wilayah AWS tanpa mengenkripsi ulang atau melakukan panggilan lintas wilayah. AWS KMS

Seperti semua kunci KMS, kunci multi-wilayah tidak pernah dibiarkan AWS KMS tidak terenkripsi. Anda dapat membuat kunci Multi-wilayah simetris atau asimetris untuk enkripsi atau penandatanganan, membuat kunci Multi-wilayah HMAC untuk membuat dan memverifikasi tag HMAC, dan membuat kunci Multi-wilayah dengan bahan kunci impor atau bahan kunci yang dihasilkan. AWS KMS Anda harus mengelola setiap kunci multi-Wilayah secara independen, termasuk membuat alias dan tag, menetapkan kebijakan dan izin kuncinya, serta mengaktifkan dan menonaktifkannya secara selektif. Anda dapat menggunakan kunci multi-Wilayah di semua operasi kriptografi yang dapat Anda lakukan dengan kunci Wilayah tunggal.

Kunci multi-Wilayah adalah solusi fleksibel dan canggih untuk berbagai skenario keamanan data umum.

Pemulihan bencana

Dalam arsitektur pencadangan dan pemulihan, kunci Multi-region memungkinkan Anda memproses data terenkripsi tanpa gangguan bahkan jika terjadi pemadaman. Wilayah AWS Data yang dikelola di Wilayah backup dapat didekripsi di Wilayah backup, dan data yang baru dienkripsi di Wilayah backup dapat didekripsi di Wilayah utama saat Wilayah tersebut dipulihkan.

Pengelolaan data global

Bisnis yang beroperasi secara global memerlukan data yang terdistribusi secara global yang tersedia secara konsisten di seluruh Wilayah AWS. Anda dapat membuat kunci multi-Wilayah di semua Wilayah di mana data Anda berada, kemudian gunakan kunci selayaknya kunci Wilayah tunggal tanpa latensi panggilan lintas Wilayah atau biaya enkripsi ulang data di berdasarkan kunci yang berbeda di setiap Wilayah.

Aplikasi penandatanganan terdistribusi

Aplikasi yang memerlukan kemampuan tanda tangan lintas Wilayah dapat menggunakan kunci penandatanganan asimetris multi-Wilayah untuk menghasilkan tanda tangan digital identik secara konsisten dan berulang kali di Wilayah AWS yang berbeda.

Jika Anda menggunakan rantai sertifikat dengan satu toko kepercayaan global (untuk satu otoritas sertifikat root (CA), dan CA perantara Regional yang ditandatangani oleh CA root, Anda tidak memerlukan kunci Multi-wilayah. Namun, jika sistem Anda tidak mendukung CA menengah, seperti penandatanganan aplikasi, Anda dapat menggunakan kunci multi-Wilayah untuk menjaga konsistensi ke sertifikasi Wilayah.

Aplikasi Active-Active yang menjangkau beberapa Wilayah

Beberapa beban kerja dan aplikasi dapat mencakup beberapa Wilayah dalam arsitektur Active-Active. Untuk aplikasi ini, kunci multi-Wilayah dapat mengurangi kompleksitas dengan menyediakan materi kunci yang sama untuk mengenkripsi dan mendekripsi operasi bersamaan pada data yang mungkin bergerak melintasi batas Wilayah.

Anda dapat menggunakan kunci multi-Wilayah dengan pustaka enkripsi di sisi klien, seperti AWS Encryption SDK, Klien Enkripsi DynamoDB, dan enkripsi di sisi klien Amazon S3. Untuk contoh penggunaan kunci Multi-wilayah dengan tabel global Amazon DynamoDB dan Klien Enkripsi DynamoDB, lihat Mengenkripsi sisi klien data global dengan kunci Multi-wilayah di Blog Keamanan. AWS KMS AWS

AWS layanan yang terintegrasi dengan AWS KMS enkripsi saat istirahat atau tanda tangan digital saat ini memperlakukan kunci Multi-wilayah seolah-olah mereka adalah kunci wilayah Tunggal. Mereka mungkin membungkus ulang atau mengenkripsi ulang data yang dipindahkan antar Wilayah. Misalnya, replikasi lintas wilayah Amazon S3 mendekripsi dan mengenkripsi ulang data di bawah kunci KMS di Wilayah tujuan, bahkan saat mereplikasi objek yang dilindungi oleh kunci Multi-wilayah.

Kunci multi-Wilayah tidak bersifat global. Anda membuat kunci primer multi-Wilayah, kemudian mereplikasi ke dalam Wilayah yang Anda pilih dalam partisi AWS. Kemudian, Anda mengelola kunci

multi-Wilayah di setiap Wilayah secara independen. Tidak ada AWS atau tidak AWS KMS pernah secara otomatis membuat atau mereplikasi kunci Multi-wilayah ke Wilayah mana pun atas nama Anda. Kunci yang dikelola AWS, kunci KMS yang dibuat AWS layanan di akun Anda untuk Anda, selalu merupakan kunci wilayah tunggal.

Anda tidak dapat mengonversi kunci Wilayah tunggal yang ada menjadi kunci multi-Wilayah. Desain ini memastikan bahwa semua data yang dilindungi dengan kunci Wilayah tunggal yang ada mempertahankan residensi data dan properti kedaulatan data yang sama.

Untuk sebagian besar kebutuhan keamanan data, isolasi Regional dan toleransi kesalahan sumber daya Regional menjadikan kunci AWS KMS Single-region standar sebagai solusi yang paling sesuai. Namun, ketika Anda perlu untuk mengenkripsi atau menandatangani data dalam aplikasi sisi klien di beberapa Wilayah, kunci multi-Wilayah mungkin solusi yang cocok.

Daerah

Tombol Multi-Region didukung di semua Wilayah AWS yang AWS KMS mendukung kecuali China (Beijing) dan China (Ningxia).

Harga dan kuota

Setiap kunci dalam satu set kunci Multi-wilayah terkait dihitung sebagai satu kunci KMS untuk harga dan kuota. AWS KMS kuota dihitung secara terpisah untuk setiap Wilayah akun. Penggunaan dan pengelolaan kunci multi-Wilayah di setiap Wilayah dihitung terhadap kuota untuk Wilayah tersebut.

Jenis kunci KMS yang didukung

Anda dapat membuat jenis kunci KMS Multi-wilayah berikut:

- Kunci KMS enkripsi simetris
- Tombol Asymmetric KMS
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor

Anda tidak dapat membuat kunci multi-Wilayah di penyimpanan kunci kustom.

Topik

- Mengontrol akses ke kunci Multi-wilayah

- Membuat kunci Multi-region
- Melihat tombol Multi-region
- Mengelola kunci Multi-region
- Mengimpor materi utama ke kunci Multi-wilayah
- Menghapus tombol Multi-region

Pertimbangan keamanan untuk kunci multi-Wilayah

Gunakan tombol AWS KMS Multi-region hanya jika Anda membutuhkannya. Kunci multi-Wilayah memberikan solusi fleksibel dan terukur untuk beban kerja yang memindahkan data terenkripsi antara Wilayah AWS atau membutuhkan akses lintas Wilayah. Pertimbangkan kunci Multi-wilayah jika Anda harus berbagi, memindahkan, atau mencadangkan data yang dilindungi di seluruh Wilayah atau perlu membuat tanda tangan digital identik dari aplikasi yang beroperasi di Wilayah yang berbeda.

Namun, proses pembuatan kunci multi-Wilayah memindahkan materi kunci Anda di batas Wilayah AWS dalam AWS KMS. Ciphertext yang dihasilkan oleh kunci multi-Wilayah berpotensi didekripsi oleh beberapa kunci terkait di beberapa lokasi geografis. Ada juga manfaat yang signifikan untuk layanan dan sumber daya yang terisolasi secara regional. Setiap Wilayah AWS terisolasi dan independen dari Wilayah lain. Wilayah memberikan toleransi kesalahan, stabilitas, dan ketahanan, dan juga dapat mengurangi latensi. Mereka memungkinkan Anda untuk membuat sumber daya berlebihan yang tetap tersedia dan tidak terpengaruh oleh pemadaman di Wilayah lain. Di AWS KMS, mereka juga memastikan bahwa setiap ciphertext dapat didekripsi hanya dengan satu kunci.

Kunci multi-Wilayah juga meningkatkan pertimbangan keamanan baru:

- Mengontrol akses dan menegakkan kebijakan keamanan data lebih kompleks dengan kunci multi-Wilayah. Anda perlu memastikan bahwa kebijakan diaudit secara konsisten pada kunci di beberapa wilayah terpencil. Anda perlu menggunakan kebijakan untuk menegakkan batasnya, bukan mengandalkan kunci terpisah.

  Misalnya, Anda perlu mengatur syarat kebijakan pada data untuk mencegah tim penggajian di satu Wilayah agar tidak dapat membaca data payroll untuk Wilayah yang berbeda. Selain itu, Anda harus menggunakan kontrol akses untuk mencegah skenario di mana kunci multi-Wilayah dalam satu Wilayah melindungi data satu penyewa dan kunci multi-Wilayah terkait di Wilayah lain melindungi data penyewa yang berbeda.

- Kunci audit di seluruh Wilayah juga lebih kompleks. Dengan kunci multi-Wilayah, Anda perlu memeriksa dan mendamaikan aktivitas audit di beberapa Wilayah untuk mendapatkan pemahaman lengkap tentang aktivitas utama pada data yang dilindungi.
- Kepatuhan terhadap mandat residensi data bisa lebih kompleks. Dengan Wilayah terisolasi, Anda dapat memastikan kepatuhan data residensi dan kedaulatan data. Kunci KMS di Wilayah tertentu dapat mendekripsi data sensitif hanya di Wilayah tersebut. Data yang dienkripsi dalam satu Wilayah dapat tetap sepenuhnya dilindungi dan tidak dapat diakses di Wilayah lain.

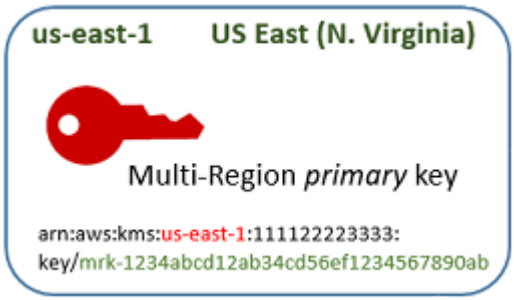
  Untuk memverifikasi residensi data dan kedaulatan data dengan kunci Multi-region, Anda perlu menerapkan kebijakan akses dan mengkompilasi peristiwa di beberapa Wilayah. AWS CloudTrail

Untuk memudahkan Anda mengelola kontrol akses pada kunci Multi-region, izin untuk mereplikasi kunci Multi-region (kms: ReplicateKey) terpisah dari izin standar untuk membuat kunci (kms:). CreateKey Selain itu, AWS KMS mendukung beberapa kondisi kebijakan untuk kunci Multi-wilayah, termasuk`kms:MultiRegion`, yang mengizinkan atau menolak izin untuk membuat, menggunakan, atau mengelola kunci Multi-wilayah dan`kms:ReplicaRegion`, yang membatasi Wilayah tempat kunci Multi-wilayah dapat direplikasi. Untuk detailnya, lihat Mengontrol akses ke kunci multi-Wilayah.

Cara kerja kunci multi-Wilayah

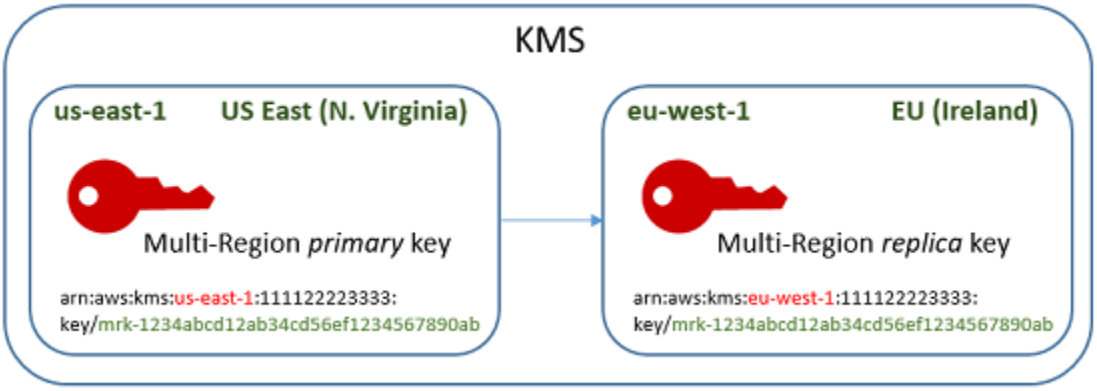
Anda mulai dengan membuat kunci primer Multi-wilayah simetris atau asimetris dalam Wilayah AWS yang AWS KMS mendukung, seperti US East (Virginia N.). Anda memutuskan apakah kunci adalah Wilayah tunggal atau multi-Wilayah hanya ketika Anda membuatnya; Anda tidak dapat mengubah properti ini nanti. Seperti halnya kunci KMS lainnya, Anda menetapkan kebijakan kunci untuk kunci Multi-region, dan Anda dapat membuat hibah, serta menambahkan alias dan tag untuk kategorisasi dan otorisasi. (Ini adalah Properti independen yang tidak dibagikan atau disinkronkan dengan kunci lainnya.) Anda dapat menggunakan kunci primer multi-Wilayah Anda dalam operasi kriptografi untuk enkripsi atau penandatanganan.

Anda dapat membuat kunci utama Multi-region di AWS KMS konsol atau dengan menggunakan CreateKeyAPI dengan `MultiRegion` parameter yang disetel ke`true`. Perhatikan bahwa kunci multi-Wilayah memiliki ID kunci khas yang dimulai dengan `mrk-`. Anda dapat menggunakan prefiks `mrk-` untuk mengidentifikasi MRK pemrograman.



Jika Anda memilih, Anda dapat mereplikasi kunci utama Multi-wilayah menjadi satu atau lebih yang berbeda Wilayah AWS di AWS partisi yang sama, seperti Eropa (Irlandia). Bila Anda melakukannya, AWS KMS buat kunci replika di Wilayah tertentu dengan ID kunci yang sama dan properti bersama lainnya sebagai kunci utama. Kemudian dengan aman mengangkut materi utama melintasi batas Wilayah dan mengaitkannya dengan kunci KMS baru di Wilayah tujuan, semuanya di dalamnya. AWS KMS Hasilnya adalah dua kunci multi-Wilayah terkait — kunci primer dan kunci replika — yang dapat digunakan secara bergantian.

Anda dapat membuat kunci replika Multi-region di AWS KMS konsol atau dengan menggunakan API. ReplicateKey



Kunci replika Multi-region yang dihasilkan adalah kunci KMS yang berfungsi penuh dengan properti bersama yang sama dengan kunci utama. Dalam semua hal lain, ini adalah kunci KMS independen dengan deskripsi, kebijakan kunci, hibah, alias, dan tag sendiri. Mengaktifkan atau menonaktifkan kunci multi-Wilayah tidak berpengaruh pada kunci multi-Wilayah terkait. Anda dapat menggunakan kunci primer dan replika secara independen dalam operasi kriptografi atau mengoordinasikan penggunaannya. Misalnya, Anda dapat mengenkripsi data dengan kunci primer di Wilayah US East (N. Virginia), memindahkan data ke Wilayah Eropa (Irlandia) dan menggunakan kunci replika untuk mendekripsi data.

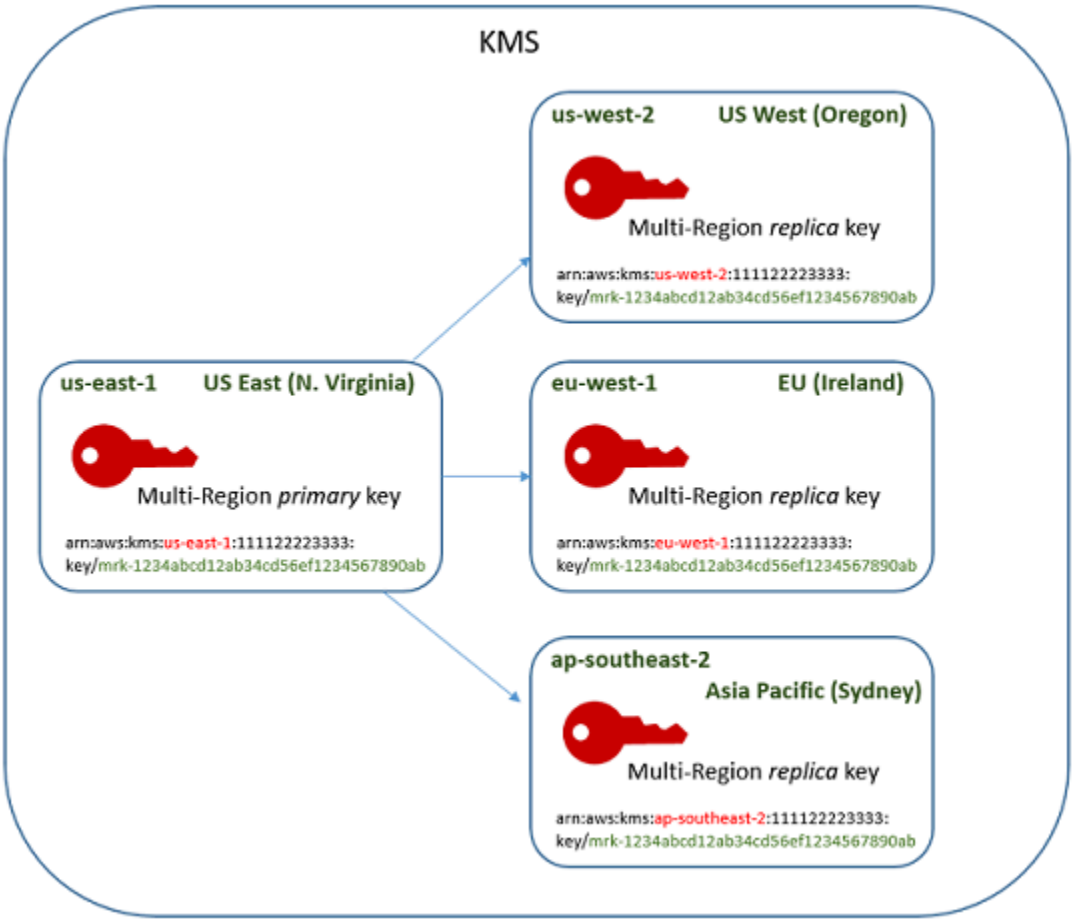
Kunci multi-Wilayah terkait memiliki ID kunci yang sama. ARN kuncinya (Nama Sumber Daya Amazon) hanya berbeda di bidang Wilayah. Misalnya, kunci primer multi-Wilayah dan kunci replika

mungkin memiliki contoh ARN kunci berikut. Kunci ID - elemen terakhir dalam ARN kunci - yang bersifat identik. Kedua kunci memiliki ID kunci khas dari kunci Multi-wilayah, yang dimulai dengan mrk-.

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

Memiliki ID kunci yang sama yang diperlukan untuk interoperabilitas. Saat mengenkripsi, AWS KMS mengikat ID kunci dari kunci KMS ke ciphertext sehingga ciphertext dapat didekripsi hanya dengan kunci KMS atau kunci KMS dengan ID kunci yang sama. Fitur ini juga membuat kunci multi-Wilayah terkait mudah dikenali, dan membuatnya lebih mudah untuk menggunakannya secara bergantian. Misalnya, ketika menggunakannya dalam aplikasi, Anda dapat merujuk ke kunci multi-Wilayah terkait dengan ID kunci bersama mereka. Kemudian, jika perlu, tentukan Wilayah atau ARN untuk membedakannya.

Saat data Anda perlu berubah, Anda dapat mereplikasi kunci utama ke yang lain Wilayah AWS di partisi yang sama, seperti US West (Oregon) dan Asia Pasifik (Sydney). Hasilnya adalah empat kunci Multi-region terkait dengan bahan kunci dan ID kunci yang sama, seperti yang ditunjukkan pada diagram berikut. Anda mengelola kunci secara terpisah. Anda dapat menggunakannya secara terpisah atau secara terkoordinasi. Misalnya, Anda dapat mengenkripsi data dengan kunci replika di Asia Pacific (Sydney), memindahkan data ke US West (Oregon), dan mendekripsi dengan kunci replika di US West (Oregon).



Pertimbangan lain untuk kunci multi-Wilayah termasuk berikut ini.

Menyinkronkan properti bersama - Jika properti bersama dari kunci Multi-region berubah, AWS KMS secara otomatis menyinkronkan perubahan dari kunci utama ke semua kunci replika. Anda tidak dapat meminta atau memaksa sinkronisasi properti bersama. AWS KMS mendeteksi dan menyinkronkan semua perubahan untuk Anda. Namun, Anda dapat mengaudit sinkronisasi dengan menggunakan SynchronizeMultiRegionKeyperistiwa di CloudTrail log.

Misalnya, jika Anda mengaktifkan rotasi tombol otomatis pada kunci primer Multi-wilayah simetris, AWS KMS salin pengaturan itu ke semua kunci replika. Ketika materi kunci diputar, rotasi disinkronkan di antara semua kunci multi-Wilayah terkait, sehingga mereka terus memiliki materi kunci saat ini yang sama, dan akses ke semua versi lama dari materi kunci. Jika Anda membuat kunci replika baru, ia memiliki bahan kunci saat ini sama dari semua kunci multi-Wilayah terkait dan akses ke semua versi sebelumnya dari materi kunci. Untuk detailnya, lihat Memutar kunci multi-Wilayah.

Mengubah kunci primer — Setiap set kunci multi-Wikayah harus memiliki tepatnya satu kunci utama. Parameter kunci primer adalah satu-satunya kunci yang dapat direplikasi. Ini juga merupakan sumber properti bersama dari kunci replika. Tapi Anda dapat mengubah kunci primer untuk replika dan mempromosikan salah satu kunci replika untuk primer. Anda dapat melakukannya sehingga Anda dapat menghapus kunci primer multi-Wilayah dari Wilayah tertentu, atau menemukan kunci utama di Wilayah yang lebih dekat dengan administrator proyek. Untuk detailnya, lihat Memperbarui Wilayah primer.

Menghapus kunci Multi-region — Seperti semua kunci KMS, Anda harus menjadwalkan penghapusan kunci Multi-region sebelum menghapusnya. AWS KMS Saat kunci menunggu penghapusan, Anda tidak dapat menggunakannya dalam operasi kriptografi apapun. Namun, tidak AWS KMS akan menghapus kunci utama Multi-wilayah sampai semua kunci replika dihapus. Lihat perinciannya di Menghapus kunci multi-Wilayah.

Konsep

Istilah dan konsep berikut digunakan dengan kunci multi-Wilayah.

Kunci multi-Wilayah

Kunci Multi-region adalah salah satu dari sekumpulan kunci KMS dengan ID kunci dan materi kunci yang sama (dan properti bersama lainnya) yang berbeda. Wilayah AWS Setiap kunci Multi-region adalah kunci KMS yang berfungsi penuh yang dapat digunakan sepenuhnya secara independen dari kunci Multi-region terkait. Karena semua kunci Multi-wilayah terkait memiliki ID kunci dan materi kunci yang sama, kunci tersebut dapat dioperasikan, yaitu, kunci Multi-wilayah terkait di mana pun Wilayah AWS dapat mendekripsi ciphertext yang dienkripsi oleh kunci Multi-wilayah terkait lainnya.

Anda mengatur properti Multi-region dari kunci KMS saat Anda membuatnya. Anda tidak dapat mengubah properti Multi-region pada kunci yang ada. Anda tidak dapat mengonversi kunci Single-region menjadi kunci Multi-region atau mengonversi kunci Multi-region menjadi kunci Single-region. Untuk memindahkan beban kerja yang ada ke dalam skenario Multi-wilayah, Anda harus mengenkripsi ulang data Anda atau membuat tanda tangan baru dengan kunci Multi-wilayah baru.

Kunci multi-wilayah dapat simetris atau asimetris dan dapat menggunakan bahan kunci atau bahan AWS KMS kunci impor. Anda tidak dapat membuat kunci multi-Wilayah di penyimpanan kunci kustom.

Dalam satu set kunci multi-Wilayah terkait, ada persis satu kunci primer pada setiap saat. Anda dapat membuat kunci replika dari kunci primer tersebut di Wilayah AWS lain. Anda juga dapat memperbarui

wilayah primer, yang mengubah kunci primer untuk kunci replika dan perubahan kunci replika tertentu untuk kunci primer. Namun, Anda hanya dapat mempertahankan satu kunci utama atau kunci replika di masing-masing Wilayah AWS. Semua Wilayah harus berada dalam partisi AWS yang sama.

Anda dapat memiliki beberapa kumpulan kunci multi-Wilayah terkait di Wilayah AWS yang sama atau berbeda. Meskipun kunci multi-Wilayah terkait interoperable, kunci multi-Wilayah yang tidak terkait tidak dapat dioperasikan.

Kunci primer

Kunci utama Multi-region adalah kunci KMS yang dapat direplikasi ke yang lain Wilayah AWS di partisi yang sama. Setiap set kunci multi-Wilayah hanya memiliki satu kunci primer.

Kunci utama berbeda dari kunci replika dengan cara berikut:

- Hanya kunci primer yang dapat direplika.
- Kunci utama adalah sumber untuk properti bersama dari replika kuncinya, termasuk materi kunci dan ID kunci.
- Anda dapat mengaktifkan dan menonaktifkan rotasi kunci otomatis hanya pada kunci primer.
- Anda dapat menjadwalkan penghapusan kunci primer pada setiap saat. Tetapi tidak AWS KMS akan menghapus kunci utama sampai semua kunci replika dihapus.

Namun, kunci primer dan replika tidak berbeda dalam properti kriptografi apa pun. Anda dapat menggunakan kunci utama dan kunci replika secara bergantian.

Anda tidak perlu mereplikasi kunci primer. Anda dapat menggunakannya seperti halnya kunci KMS apa pun dan mereplikasi jika dan kapan itu berguna. Namun, karena kunci Multi-region memiliki properti keamanan yang berbeda dari kunci Single-region, sebaiknya Anda membuat kunci Multi-region hanya jika Anda berencana untuk mereplikasi kunci tersebut.

Kunci replika

Kunci replika Multi-region adalah kunci KMS yang memiliki ID kunci dan materi kunci yang sama dengan kunci utama dan kunci replika terkait, tetapi ada dalam kunci yang berbeda. Wilayah AWS

Kunci replika adalah kunci KMS yang berfungsi penuh dengan kebijakan kunci, hibah, alias, tag, dan properti lainnya sendiri. Ini bukan salinan atau pointer ke kunci primer atau kunci lainnya. Anda dapat menggunakan kunci replika bahkan jika kunci primer dan semua kunci replika terkait dinonaktifkan. Anda juga dapat mengkonversi kunci replika untuk kunci primer dan kunci primer untuk

kunci replika. Setelah dibuat, kunci replika bergantung pada kunci primernya hanya untuk rotasi kunci dan memperbarui Wilayah primer.

Kunci primer dan replika tidak berbeda dalam properti kriptografi apa pun. Anda dapat menggunakan kunci utama dan kunci replika secara bergantian. Data yang dienkripsi oleh kunci primer atau replika dapat didekripsi dengan kunci yang sama, atau oleh kunci primer atau replika terkait.

Replikasi

Anda dapat mereplikasi kunci utama Multi-region menjadi yang berbeda Wilayah AWS di partisi yang sama. Bila Anda melakukannya, AWS KMS buat kunci replika Multi-region di Region yang ditentukan dengan ID kunci yang sama dan properti bersama lainnya sebagai kunci utamanya. Kemudian, kunci dengan aman mengirim materi kunci di seluruh batas Wilayah dan mengaitkannya dengan kunci replika baru di Wilayah tujuan, semua dalam AWS KMS.

Properti bersama

Properti bersama adalah properti dari kunci primer Multi-wilayah yang dibagikan dengan kunci replika. AWS KMS membuat kunci replika dengan nilai properti bersama yang sama dengan kunci utama. Kemudian, secara berkala menyinkronkan nilai properti bersama kunci primer untuk kunci replika. Anda tidak dapat mengatur properti ini pada kunci replika.

Berikut ini adalah properti bersama dari kunci multi-Wilayah.

- ID Kunci — (Elemen `Region` dari ARN kunci berbeda.)
- Bahan kunci
- Asal bahan utama
- Spesifikasi kunci dan algoritme enkripsi
- Penggunaan kunci
- Rotasi kunci otomatis — Anda dapat mengaktifkan dan menonaktifkan rotasi kunci otomatis hanya pada kunci primer. Kunci replika baru dibuat dengan semua versi dari materi kunci bersama. Untuk detailnya, lihat Memutar kunci multi-Wilayah.
- Rotasi sesuai permintaan — Anda dapat melakukan rotasi sesuai permintaan hanya pada kunci utama. Kunci replika baru dibuat dengan semua versi dari materi kunci bersama. Untuk detailnya, lihat Memutar kunci multi-Wilayah.

Anda juga dapat memikirkan sebutan primer dan replika kunci multi-Wilayah terkait sebagai properti bersama. Saat Anda membuat kunci replika baru atau memperbarui kunci utama, AWS KMS

menyinkronkan perubahan ke semua kunci Multi-wilayah terkait. Ketika perubahan ini selesai, semua kunci multi-Wilayah terkait mencantumkan kunci primer dan replikanya secara akurat.

Semua properti lain dari kunci multi-Wilayah adalah properti independen, termasuk deskripsi, kebijakan kunci, izin, status kunci diaktifkan dan dinonaktifkan, alias, dan tag. Anda dapat mengatur nilai yang sama untuk properti ini pada semua kunci multi-Wilayah terkait, tetapi jika Anda mengubah nilai properti independen, AWS KMS tidak menyinkronkannya.

Anda dapat melacak sinkronisasi properti bersama kunci multi-Wilayah Anda. Di AWS CloudTrail log Anda, cari SynchronizeMultiRegionKeyacara tersebut.

Mengontrol akses ke kunci multi-Wilayah

Anda dapat menggunakan kunci multi-Wilayah sesuai dengan kepatuhan, pemulihan bencana, dan skenario backup yang akan lebih kompleks dengan kunci Wilayah tunggal. Namun, karena properti keamanan dari kunci multi-wilayah secara signifikan berbeda dengan kunci Wilayah tunggal, kami merekomendasikan sebaiknya berhati-hati saat mengotorisasi pembuatan, manajemen, dan penggunaan kunci multi-Wilayah.

> **Note**
>
> Pernyataan kebijakan IAM yang ada dengan karakter wildcard di `Resource` bidang sekarang berlaku untuk kunci Single-region dan Multi-region. Untuk membatasi mereka ke kunci KMS wilayah tunggal atau kunci Multi-wilayah, gunakan kunci kondisi kms:. MultiRegion

Gunakan alat otorisasi Anda untuk mencegah pembuatan dan penggunaan kunci multi-Wilayah dalam skenario apa pun di mana Wilayah tunggal saja cukup. Izinkan perwakilan untuk mereplikasi kunci multi-Wilayah hanya keWilayah AWS yang membutuhkannya. Berikan izin kunci multi-Wilayah hanya kepada perwakilan yang membutuhkan dan hanya untuk tugas-tugas yang memerlukannya.

Anda dapat menggunakan kebijakan kunci, kebijakan IAM, dan izin untuk mengizinkan perwakilan IAM mengelola dan menggunakan kunci multi-Wilayah di Akun AWS Anda. Setiap kunci multi-Wilayah adalah sumber daya independen dengan ARN dan kebijakan kunci yang unik. Anda perlu menetapkan dan memelihara kebijakan kunci untuk setiap kunci dan memastikan bahwa kebijakan IAM baru maupun yang sudah ada menerapkan strategi otorisasi Anda.

Topik

- Basic otorisasi untuk kunci multi-Wilayah

- Mengotorisasi administrator dan pengguna kunci multi-Wilayah
- Mengotorisasi AWS KMS untuk menyinkronkan kunci multi-Wilayah

Basic otorisasi untuk kunci multi-Wilayah

Ketika merancang kebijakan kunci dan kebijakan IAM untuk kunci multi-Wilayah, pertimbangkan prinsip-prinsip berikut.

- Kebijakan utama - Setiap kunci Multi-wilayah adalah sumber daya kunci KMS independen dengan kebijakan utamanya sendiri. Anda dapat menerapkan kebijakan kunci yang sama maupun berbeda untuk setiap kunci dari kumpulan kunci multi-Wilayah terkait. Kebijakan kunci bukan properti bersama dari kunci multi-Wilayah. AWS KMS tidak menyalin atau menyinkronkan kebijakan kunci di antara kunci multi-Wilayah terkait.

  Ketika Anda membuat kunci replika di konsol AWS KMS, konsol tersebut menampilkan kebijakan kunci dari kunci primer saat ini sebagai bentuk kenyamanan. Anda dapat menggunakan kebijakan kunci ini, mengeditnya, atau menghapus serta menggantinya. Tetapi bahkan jika Anda menerima kebijakan kunci primer yang tidak berubah, AWS KMS tidak menyinkronkan kebijakan. Sebagai contoh, jika Anda mengubah kebijakan kunci dari kunci primer, kebijakan kunci dari replika tetap sama.
- Kebijakan kunci default — Saat Anda membuat kunci Multi-wilayah dengan menggunakan CreateKeydan `ReplicateKey` operasi, kebijakan kunci default diterapkan kecuali Anda menentukan kebijakan kunci dalam permintaan. Ini adalah kebijakan kunci default yang sama yang diterapkan untuk kunci wilayah tunggal.
- Kebijakan IAM — Seperti semua kunci KMS, Anda dapat menggunakan kebijakan IAM untuk mengontrol akses ke kunci Multi-wilayah hanya jika kebijakan kunci mengizinkannya. Kebijakan IAM berlaku untuk semua Wilayah AWS secara default. Namun, Anda dapat menggunakan kunci kondisi, seperti aws: RequestedRegion, untuk membatasi izin ke Wilayah tertentu.

  Untuk membuat kunci primer dan replika, perwakilan harus memiliki izin `kms:CreateKey` dalam kebijakan IAM yang berlaku untuk wilayah di mana kunci dibuat.
- Izin — Izin AWS KMS bersifat Regional. Setiap hibah memungkinkan izin untuk satu kunci KMS. Anda dapat menggunakan izin untuk mengizinkan izin untuk kunci primer multi-wilayah atau kunci replika. Tetapi Anda tidak dapat menggunakan satu hibah untuk mengizinkan izin ke beberapa kunci KMS, bahkan jika itu adalah kunci Multi-wilayah terkait.

- ARN kunci — Setiap kunci multi-Wilayah memiliki ARN kunci unik. Kunci ARN dari kunci multi-Wilayah terkait memiliki partisi, akun, dan ID kunci, tetapi dengan Wilayah yang berbeda.

  Guna menerapkan pernyataan kebijakan IAM untuk kunci multi-Wilayah tertentu, gunakan ARN kunci atau pola ARN kuncinya yang mencakup Wilayah. Untuk menerapkan pernyataan kebijakan IAM untuk semua kunci multi-wilayah terkait, menggunakan karakter kartubebas (*) dalam elemen Wilayah dari ARN, seperti yang ditunjukkan dalam contoh berikut.

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

  Untuk menerapkan pernyataan kebijakan ke semua kunci Multi-wilayah di AndaAkun AWS, Anda dapat menggunakan kondisi MultiRegion kebijakan kms: atau pola ID kunci yang menyertakan awalan khususmrk-.

- Peran terkait layanan — Kepala sekolah yang membuat kunci utama Multi-wilayah harus memiliki izin iam:. CreateServiceLinkedRole

  Untuk menyinkronkan properti bersama dari kunci multi-Wilayah terkait, AWS KMS mengasumsikan peran yang tertaut ke layanan IAM. AWS KMS membuat peran yang tertaut ke layanan di Akun AWS setiap kali Anda membuat kunci primer multi-wilayah. (Jika perannya ada, AWS KMS akan membuatnya ulang, yang tidak memiliki efek berbahaya.) Peran ini berlaku di semua Wilayah. AWS KMSUntuk memungkinkan membuat (atau membuat ulang) peran terkait layanan, kepala sekolah yang membuat kunci utama Multi-wilayah harus memiliki izin iam:. CreateServiceLinkedRole

Mengotorisasi administrator dan pengguna kunci multi-Wilayah

Perwakilan yang membuat dan mengelola kunci multi-Wilayah memerlukan izin berikut di daerah primer dan replika:

- `kms:CreateKey`

- `kms:ReplicateKey`
- `kms:UpdatePrimaryRegion`
- `iam:CreateServiceLinkedRole`

Membuat kunci primer

Untuk membuat kunci primer Multi-wilayah, prinsipal memerlukan CreateServiceLinkedRole izin kms: CreateKey dan iam: dalam kebijakan IAM yang efektif di Region kunci primer. Prinsipal yang memiliki izin ini dapat membuat kunci Single-region dan Multi-region kecuali Anda membatasi izinnya.

`iam:CreateServiceLinkedRole`Izin memungkinkan AWS KMS untuk membuat AWSServiceRoleForKeyManagementServiceMultiRegionKeysperan untuk menyinkronkan properti bersama kunci Multi-wilayah terkait.

Misalnya, kebijakan IAM ini memungkinkan prinsipal untuk membuat semua jenis kunci KMS.

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

Untuk mengizinkan atau menolak izin membuat kunci utama Multi-wilayah, gunakan kunci MultiRegion kondisi kms:. Nilai yang valid adalah `true` (kunci multi-Wilayah) atau `false` (kunci Wilayah tunggal). Misalnya, pernyataan kebijakan IAM berikut menggunakan tindakan `Deny` dengan kunci syarat `kms:MultiRegion` untuk mencegah perwakilan membuat kunci multi-Wilayah.

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action":"kms:CreateKey",
      "Effect":"Deny",
      "Resource":"*",
      "Condition": {
```

```
            "Bool": "kms:MultiRegion": true
        }
    }
}
```

Mereplikasi kunci

Untuk membuat kunci replika multi-Wilayah, perwakilan memerlukan izin berikut:

- kms: ReplicateKey izin dalam kebijakan kunci kunci utama.
- kms: CreateKey izin dalam kebijakan IAM yang efektif di Region kunci replika.

Berhati-hatilah saat mengizinkan izin ini. Mereka memungkinkan prinsipal untuk membuat kunci KMS dan kebijakan utama yang mengotorisasi penggunaannya. Parameter izin `kms:ReplicateKey` juga mengizinkan transfer materi kunci di seluruh batas-batas Wilayah dalam AWS KMS.

Untuk membatasi Wilayah AWS di mana kunci Multi-region dapat direplikasi, gunakan kms: condition key. ReplicaRegion Kunci ini hanya akan membatasi izin `kms:ReplicateKey`. Jika tidak, kunci tidak berpengaruh. Misalnya, kebijakan kunci berikut mengizinkan perwakilan untuk mereplikasi kunci primer ini, tetapi hanya di Wilayah tertentu.

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

Memperbarui Wilayah primer

Perwakilan terotorisasi dapat mengkonversi kunci replika untuk kunci primer, yang mengubah kunci primer sebelumnya ke replika. Tindakan ini dikenal sebagai memperbarui Wilayah primer. Untuk memperbarui Wilayah utama, kepala sekolah membutuhkan kms: UpdatePrimaryRegion izin di kedua Wilayah. Anda dapat memberikan izin ini di kebijakan kunci atau IAM.

- `kms:UpdatePrimaryRegion` pada kunci primer. Izin ini harus efektif di Wilayah kunci primer.
- `kms:UpdatePrimaryRegion` pada kunci replika. Izin ini harus berlaku di Wilayah kunci replika.

Misalnya, kebijakan kunci berikut memberi pengguna yang dapat mengasumsikan izin peran Administrator untuk memperbarui Wilayah utama kunci KMS. Kunci KMS ini dapat menjadi kunci utama atau kunci replika dalam operasi ini.

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

Untuk membatasi kunci Wilayah AWS yang dapat meng-host kunci utama, gunakan kms: PrimaryRegion condition key. Misalnya, pernyataan kebijakan IAM berikut ini mengizinkan perwakilan untuk memperbarui Wilayah primer kunci multi-Wilayah di Akun AWS, tetapi hanya ketika Wilayah utama baru adalah salah satu dari Wilayah yang ditentukan.

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
      "arn:aws:kms:*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-west-2",
         "sa-east-1",
         "ap-southeast-1"
```

```
      ]
    }
  }
}
```

Menggunakan dan mengelola kunci multi-Wilayah

Secara default, kepala sekolah yang memiliki izin untuk menggunakan dan mengelola kunci KMS di dan Wilayah juga memiliki izin untuk menggunakan Akun AWS dan mengelola kunci Multi-wilayah. Namun, Anda dapat menggunakan kms: MultiRegion condition key untuk mengizinkan hanya kunci Single-region atau hanya kunci Multi-region. Atau gunakan kunci MultiRegionKeyType kondisi kms: untuk mengizinkan hanya kunci utama Multi-wilayah atau hanya kunci replika. Kedua tombol kondisi mengontrol akses ke CreateKeyoperasi dan operasi apa pun yang menggunakan kunci KMS yang ada, seperti Enkripsi atau. EnableKey

Contoh pernyataan kebijakan IAM berikut menggunakan kunci syarat `kms:MultiRegion` untuk mencegah perwakilan menggunakan atau mengelola kunci multi-Wilayah apa pun.

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

Contoh pernyataan kebijakan IAM berikut menggunakan kunci syarat `kms:MultiRegionKeyType` untuk mengizinkan perwakilan menjadwalkan dan membatalkan penghapusan kunci, tetapi hanya pada kunci replika multi-Wilayah.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
      "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
```

```
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Mengotorisasi AWS KMS untuk menyinkronkan kunci multi-Wilayah

Untuk mensuport kunci multi-Wilayah, AWS KMS menggunakan peran terkait layanan IAM. Peran ini memberi AWS KMS izin yang dibutuhkan untuk menyinkronkan properti bersama. Anda dapat melihat SynchronizeMultiRegionKey CloudTrail peristiwa yang merekam AWS KMS sinkronisasi properti bersama di AWS CloudTrail log Anda.

Tentang peran yang tertaut ke layanan untuk kunci multi-Wilayah

Peran yang tertaut ke layanan adalah IAM role yang memberikan satu izin layanan AWS untuk memanggil layanan AWS lainnya atas nama Anda. Ini dirancang agar mempermudah Anda menggunakan fitur dari beberapa layanan AWS terintegrasi tanpa harus membuat dan memelihara kebijakan IAM yang kompleks.

Untuk kunci Multi-wilayah, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan dengan kebijakan. AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy Kebijakan ini memberi peran izin `kms:SynchronizeMultiRegionKey`, yang mengizinkannya untuk menyinkronkan properti bersama dari kunci multi-wilayah.

Karena peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan hanya mempercayai`mrk.kms.amazonaws.com`, hanya AWS KMS dapat mengambil peran terkait layanan ini. Peran ini terbatas pada operasi yang memerlukan AWS KMS menyinkronkan properti bersama multi-wilayah. Ini tidak memberikan AWS KMS izin tambahan apa pun. Misalnya, AWS KMS tidak memiliki izin untuk membuat, mereplikasi, atau menghapus kunci KMS apa pun.

Untuk informasi lebih lanjut tentang cara layanan AWS menggunakan peran tertaut layanan, lihat Menggunakan Peran Tertaut Layanan dalam Panduan Pengguna IAM.

Membuat peran yang ditautkan ke layanan

AWS KMSsecara otomatis membuat peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan di Akun AWS saat Anda membuat kunci Multi-wilayah, jika peran tersebut belum ada. Anda tidak dapat membuat atau membuat ulang peran yang tertaut dengan layanan ini secara langsung.

Mengedit deskripsi peran tertaut layanan

Anda tidak dapat mengedit nama peran atau pernyataan kebijakan dalam peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan, tetapi Anda dapat mengedit deskripsi peran. Untuk informasi lebih lanjut, lihat Mengedit Peran Tertaut Layanan dalam Panduan Pengguna IAM.

Menghapus peran tertaut layanan

AWS KMStidak menghapus peran AWSServiceRoleForKeyManagementServiceMultiRegionKeysterkait layanan dari Anda Akun AWS dan Anda tidak dapat menghapusnya. Namun, AWS KMS tidak mengambil AWSServiceRoleForKeyManagementServiceMultiRegionKeysperan atau menggunakan salah satu izinnya kecuali Anda memiliki kunci Multi-wilayah di Akun AWS dan Wilayah Anda.

Membuat kunci multi-Wilayah

Anda dapat membuat kunci multi-Wilayah di konsol tersebut atau menggunakan API AWS KMS.

Properti Multi-region yang Anda tetapkan dalam prosedur ini tidak dapat diubah. Anda tidak dapat mengonversi kunci Single-region menjadi kunci Multi-region atau mengonversi kunci Multi-region menjadi kunci Single-region.

Topik

- Membuat kunci primer multi-Wilayah
- Membuat kunci replika multi-Wilayah

Membuat kunci primer multi-Wilayah

Anda dapat membuat kunci primer multi-Wilayah di konsol AWS KMS tersebut atau menggunakan API AWS KMS. Anda dapat membuat kunci primer dalam Wilayah AWS apa pun di mana AWS KMS mendukung kunci multi-Wilayah.

Untuk membuat kunci primer Multi-wilayah, prinsipal memerlukan izin yang sama dengan yang mereka perlukan untuk membuat kunci KMS apa pun, termasuk CreateKey izin kms: dalam kebijakan IAM. Kepala sekolah juga membutuhkan iam: CreateServiceLinkedRole izin. Anda dapat menggunakan kms: MultiRegionKeyType condition key untuk mengizinkan atau menolak izin untuk membuat kunci utama Multi-region.

Instruksi ini membuat kunci primer multi-Wilayah dengan materi kunci yang dihasilkan oleh AWS KMS. Untuk membuat kunci primer multi-Wilayah dengan materi kunci yang diimpor, lihat Membuat kunci primer dengan materi kunci yang diimpor.

Topik

- Membuat kunci primer multi-Wilayah (konsol)
- Membuat kunci primer multi-Wilayah (API AWS KMS)

Membuat kunci primer multi-Wilayah (konsol)

Untuk membuat kunci utama Multi-region di AWS KMS konsol, gunakan proses yang sama yang akan Anda gunakan untuk membuat kunci KMS apa pun.. Anda memilih kunci multi-Wilayah di Opsi lanjutan. Untuk instruksi selengkapnya, lihat Membuat kunci.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Pilih jenis kunci simetris atau asimetris. Tombol simetris adalah default.

   Anda dapat membuat kunci simetris dan asimetris multi-wilayah, termasuk kunci KMS HMAC Multi-wilayah, yang simetris.
6. Pilih penggunaan kunci Anda. Enkripsi dan dekripsi adalah default.

   Untuk bantuan, lihatthe section called “Membuat kunci”,the section called “Membuat tombol KMS asimetris”, atauthe section called “Membuat kunci HMAC”.
7. Memperluas Opsi lanjutan.

8. Di bawah Asal materi kunci, agar AWS KMS menghasilkan materi kunci yang akan dibagikan kunci utama dan replika Anda, pilih KMS. Jika Anda mengimpor materi kunci ke kunci primer dan replika, pilih Eksternal (Impor bahan kunci).
9. Di bawah Replikasi multi-Wilayah, pilih Mengizinkan kunci ini untuk direplikasi ke Wilayah lain.

   Anda tidak dapat mengubah pengaturan ini setelah Anda membuat kunci KMS.
10. Ketik alias untuk kunci utama.

    Alias bukan properti bersama kunci Multi-wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika alias yang sama atau alias yang berbeda. AWS KMStidak menyinkronkan alias kunci Multi-wilayah.

    > **Note**
    >
    > Menambahkan, menghapus, atau memperbarui alias dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan alias untuk mengontrol akses ke tombol KMS.
11. (Opsional) Ketik deskripsi kunci utama.

    Deskripsi bukan properti bersama kunci Multi-wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika deskripsi yang sama atau deskripsi yang berbeda. AWS KMStidak menyinkronkan deskripsi kunci kunci Multi-region.
12. (Opsional) Ketik kunci tanda dan nilai tanda opsional. Untuk menetapkan lebih dari satu tag ke kunci utama, pilih Tambah tag.

    Tag bukanlah properti bersama dari kunci multi-Wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika tag yang sama atau tag yang berbeda. AWS KMStidak menyinkronkan tag kunci Multi-wilayah. Anda dapat mengubah tag pada tombol KMS kapan saja.

    > **Note**
    >
    > Menandai atau melepas tag kunci KMS dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.
13. Pilih pengguna IAM dan peran yang dapat mengelola kunci utama.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk mengelola kunci KMS.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Langkah ini memulai proses pembuatan kebijakan kunci untuk kunci utama. Kebijakan utama bukan properti bersama kunci Multi-wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika kebijakan kunci yang sama atau kebijakan kunci yang berbeda. AWS KMStidak menyinkronkan kebijakan kunci kunci Multi-region. Anda dapat mengubah kebijakan kunci kunci KMS kapan saja.

14. Selesaikan langkah-langkah untuk membuat kebijakan utama, termasuk memilih pengguna utama. Setelah Anda meninjau kebijakan kunci, pilih Selesai untuk membuat kunci KMS.

Membuat kunci primer multi-Wilayah (API AWS KMS)

Untuk membuat kunci primer Multi-wilayah, gunakan CreateKeyoperasi. Gunakan parameter `MultiRegion` dengan nilai `True`.

Misalnya, perintah berikut membuat kunci primer multi-Wegion di pemanggilWilayah AWS (us-east-1). Ia menerima nilai default untuk semua properti lainnya, termasuk kebijakan kunci. Nilai default untuk kunci primer Multi-region sama dengan nilai default untuk semua kunci KMS lainnya, termasuk kebijakan kunci default. Prosedur ini menciptakan kunci enkripsi simetris, kunci KMS default.

Tanggapan meliputi elemen `MultiRegion` dan elemen `MultiRegionConfiguration` dengan sub-elemen khas dan nilai untuk kunci primer multi-Wilayah tanpa kunci replika. ID kunci dari kunci multi-wilayah selalu dimulai dengan `mrk-`.

Important

Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

```
$ 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": [ ]
      }
    }
}
```

Membuat kunci replika multi-Wilayah

Anda dapat membuat kunci replika Multi-region di AWS KMS konsol, dengan menggunakan ReplicateKeyoperasi, atau dengan menggunakan template. AWS CloudFormation Anda tidak dapat menggunakan CreateKeyoperasi untuk membuat kunci replika.

Anda dapat menggunakan prosedur ini untuk mereplikasi kunci utama Multi-region, termasuk kunci KMS enkripsi simetris, kunci KMSasimetris, atau kunci KMS HMAC.

Ketika operasi ini selesai, kunci replika baru memiliki status kunci sementara dari `Creating`. Status kunci ini berubah menjadi `Enabled` (atau PendingImport) setelah beberapa detik ketika proses pembuatan kunci replika baru selesai. Sementara status kunci adalah `Creating`, Anda dapat mengelola kunci, tetapi Anda belum bisa menggunakannya dalam operasi kriptografi. Jika Anda membuat dan menggunakan kunci replika secara terprogram, coba lagi `KMSInvalidStateException` atau panggil DescribeKeyuntuk memeriksa `KeyState` nilainya sebelum menggunakannya.

Jika Anda salah menghapus kunci replika, Anda dapat menggunakan prosedur ini untuk membuatnya kembali. Jika Anda mereplikasi kunci utama yang sama di Wilayah yang sama, kunci replika baru yang Anda buat akan memiliki properti bersama yang sama dengan kunci replika asli.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Pelajari selengkapnya

- Untuk membuat kunci replika multi-Wilayah dengan materi kunci yang diimpor, lihat Membuat kunci replika dengan materi kunci yang diimpor.
- Untuk menggunakan AWS CloudFormation template untuk membuat kunci replika, lihat AWS::KMS::ReplicaKeydi Panduan AWS CloudFormation Pengguna.

Topik

- Wilayah Replika
- Membuat kunci replika (konsol)
- Membuat kunci replika (API AWS KMS)

Wilayah Replika

Anda biasanya memilih untuk mereplikasi kunci multi-Wilayah ke Wilayah AWSberdasarkan model bisnis dan persyaratan peraturan Anda. Misalnya, Anda mungkin mereplikasi kunci ke Wilayah di mana Anda menyimpan sumber daya Anda. Atau, untuk memenuhi persyaratan pemulihan bencana, Anda mungkin mereplikasi kunci ke Wilayah yang jauh secara geografis.

Berikut ini adalah persyaratan AWS KMS untuk replika Wilayah. Jika Wilayah yang Anda pilih tidak sesuai dengan persyaratan ini, upaya untuk mereplikasi kunci gagal.

- Satu kunci multi-wilayah terkait per Wilayah — Anda tidak dapat membuat kunci replika di Wilayah yang sama dengan kunci primernya, atau di Wilayah yang sama dengan replika kunci primer lainnya.

  Jika Anda mencoba mereplikasi kunci utama di Wilayah yang sudah memiliki replika kunci utama itu, upaya gagal. Jika kunci replika saat ini di Wilayah berada dalam status `PendingDeletion` kunci, Anda dapat membatalkan penghapusan kunci replika atau menunggu hingga kunci replika dihapus.
- Beberapa kunci multi-wilayah yang tidak terkait di Wilayah yang sama — Anda dapat memiliki beberapa kunci multi-Wilah yang tidak terkait di Wilayah yang sama. Misalnya, Anda dapat memiliki dua kunci primer multi-Wilayah di Wilayah `us-east-1`. Masing-masing kunci primer dapat memiliki kunci replika di Wilayah `us-west-2`.
- Wilayah di partisi yang sama — Kunci replika Wilayah harus berada di partisi AWS yang sama dengan kunci primer Wilayah.
- Wilayah harus diaktifkan — Jika Wilayah dinonaktifkan secara default, Anda tidak dapat membuat sumber daya apa pun di Wilayah tersebut sampai Wilayah diaktifkan untuk Akun AWS Anda.

Membuat kunci replika (konsol)

Di konsol AWS KMS tersebut, Anda dapat membuat satu atau banyak replika kunci primer multi-Wilayah dalam operasi yang sama.

Prosedur ini mirip dengan membuat kunci KMS wilayah tunggal standar di konsol. Namun, karena kunci replika didasarkan pada kunci primer, Anda tidak memilih nilai untuk properti bersama, seperti spesifikasi kunci (simetris atau asimetris), penggunaan kunci, atau asal kunci.

Anda menentukan properti yang tidak dibagi, termasuk alias, tanda, deskripsi, dan kebijakan kunci. Sebagai kenyamanan, konsol tersebut menampilkan nilai properti saat ini dari kunci primer, namun

Anda dapat mengubahnya. Bahkan jika Anda menyimpan nilai-nilai kunci primer, AWS KMS tidak menyinkronkan nilai ini.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih ID kunci atau alias dari kunci primer multi-Wilayah. Ini membuka halaman detail kunci untuk kunci KMS.

   Untuk mengidentifikasi kunci primer multi-Wilayah, gunakan ikon alat di sudut kanan atas untuk menambahkan kolom Regionalitas ke tabel.
5. Pilih tab Regionalitas.
6. Pada bagian Kunci multi-Wilayah yang terkait, pilih Buat kunci replika baru.

   Bagian Kunci multi-Wilayah yang terkait menampilkan wilayah kunci primer dan kunci replika. Anda dapat menggunakan tampilan ini untuk membantu Anda memilih Wilayah untuk kunci replika baru Anda.
7. Pilih satu atau lebih Wilayah AWS. Prosedur ini membuat kunci replika di masing-masing Wilayah yang Anda pilih.

   Menu hanya mencakup Wilayah di partisi AWS yang sama sebagai kunci primer. Wilayah yang sudah memiliki kunci multi-Wilayah terkait ditampilkan, tetapi tidak dapat dipilih. Anda mungkin tidak memiliki izin untuk mereplikasi kunci ke semua Wilayah pada menu.

   Setelah selesai memilih Wilayah, tutup menu. Wilayah yang Anda pilih akan ditampilkan. Untuk membatalkan replikasi ke Wilayah, pilih X di samping nama Wilayah.
8. Ketik alias untuk kunci replika.

   Konsol tersebut menampilkan salah satu alias kunci primer saat ini, namun Anda dapat mengubahnya. Anda dapat memberikan kunci utama Multi-wilayah dan replika alias yang sama

atau alias yang berbeda. Alias bukan properti bersama kunci Multi-wilayah. AWS KMStidak menyinkronkan alias kunci Multi-wilayah.

Menambahkan, menghapus, atau memperbarui alias dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan alias untuk mengontrol akses ke tombol KMS.

9. (Opsional) Ketik deskripsi kunci replika.

   Konsol tersebut menampilkan deskripsi kunci primer saat ini, namun Anda dapat mengubahnya. Deskripsi bukan properti bersama kunci Multi-wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika deskripsi yang sama atau deskripsi yang berbeda. AWS KMStidak menyinkronkan deskripsi kunci kunci Multi-region.

10. (Opsional) Ketik kunci tanda dan nilai tanda opsional. Untuk menetapkan lebih dari satu tag ke kunci replika, pilih Tambah tag.

    Konsol tersebut menampilkan tanda yang saat ini terlampir pada kunci primer, tetapi Anda dapat mengubahnya. Tag bukanlah properti bersama dari kunci multi-Wilayah. Anda dapat memberikan kunci utama Multi-wilayah dan replika tag yang sama atau tag yang berbeda. AWS KMStidak menyinkronkan tag tombol Multi-wilayah.

    Menandai atau melepas tag kunci KMS dapat mengizinkan atau menolak izin ke kunci KMS. Untuk detailnya, lihat ABAC untuk AWS KMS dan Menggunakan tag untuk mengontrol akses ke tombol KMS.

11. Pilih pengguna IAM dan peran yang dapat mengelola kunci replika.

    > **Note**
    >
    > Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk mengelola kunci replika.
    > Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

    Langkah ini memulai proses pembuatan kebijakan kunci untuk kunci replika. Konsol tersebut menampilkan kebijakan kunci primer saat ini, namun Anda dapat mengubahnya. Kebijakan utama bukan properti bersama kunci Multi-wilayah. Anda dapat memberikan kunci utama

Multi-wilayah dan replika kebijakan kunci yang sama atau kebijakan kunci yang berbeda. AWS KMStidak menyinkronkan kebijakan utama. Anda dapat mengubah kebijakan kunci kunci KMS kapan saja.

12. Selesaikan langkah-langkah untuk membuat kebijakan utama, termasuk memilih pengguna utama. Setelah Anda meninjau kebijakan kunci, pilih Selesai untuk membuat kunci replika.

Membuat kunci replika (API AWS KMS)

Untuk membuat kunci replika Multi-wilayah, gunakan operasi. ReplicateKey Anda tidak dapat menggunakan CreateKeyoperasi untuk membuat kunci replika. Operasi ini menciptakan satu kunci replika pada satu waktu. Wilayah yang Anda tentukan harus mematuhi Persyaratan wilayah untuk kunci replika.

Ketika Anda menggunakan operasi `ReplicateKey`, Anda tidak menentukan nilai untuk properti bersama kunci multi-Wilayah apa pun. Nilai properti bersama disalin dari kunci primer dan terus disinkronkan. Namun, Anda dapat menentukan nilai untuk properti yang tidak dibagi. Jika tidak, AWS KMS menerapkan nilai default standar untuk kunci KMS, bukan nilai kunci primer.

> **Note**
>
> Jika Anda tidak menentukan nilai untuk`Description`,`KeyPolicy`, atau `Tags` parameter, AWS KMS membuat kunci replika dengan deskripsi string kosong, kebijakan kunci default, dan tanpa tag.
> Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Misalnya, perintah berikut membuat kunci replika multi-Wilayah di Wilayah Asia Pacific (Sydney) (ap-southeast-2). Kunci replika ini dimodelkan pada kunci primer di Wilayah US East (N. Virginia) (us-east-1), yang diidentifikasi oleh nilai parameter `KeyId`. Contoh ini menerima nilai default untuk semua properti lainnya, termasuk kebijakan kunci.

Tanggapan menjelaskan kunci replika baru. Ini mencakup bidang untuk properti bersama, seperti`KeyId`,, `KeySpec``KeyUsage`, dan key material origin (`Origin`). Ini juga mencakup properti yang independen dari kunci primer, seperti `Description`, kebijakan kunci (`ReplicaKeyPolicy`), dan tanda (`ReplicaTags`).

Tanggapan juga mencakup ARN kunci dan wilayah kunci primer serta semua kunci replikanya, termasuk salah satu yang baru saja dibuat di Wilayah ap-southeast-2. Dalam contoh ini, elemen `ReplicaKey` menunjukkan bahwa kunci primer ini sudah direplikasi di Wilayah Eropa (Irlandia) (eu-west-1).

```
$ aws kms replicate-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
    --replica-region ap-southeast-2
{
    "ReplicaKeyMetadata": {
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "REPLICA",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                },
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                }
            ]
        },
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1607472987.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
```

```
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    },
    "ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-
default-1\",...,
    "ReplicaTags": []
}
```

Melihat kunci multi-Wilayah

Anda dapat melihat kunci Wilayah tunggal dan multi-Wilayah di konsol AWS KMS tersebut serta menggunakan operasi API AWS KMS.

Topik

- Melihat kunci multi-Wilayah di konsol tersebut
- Melihat kunci multi-Wilayah di API

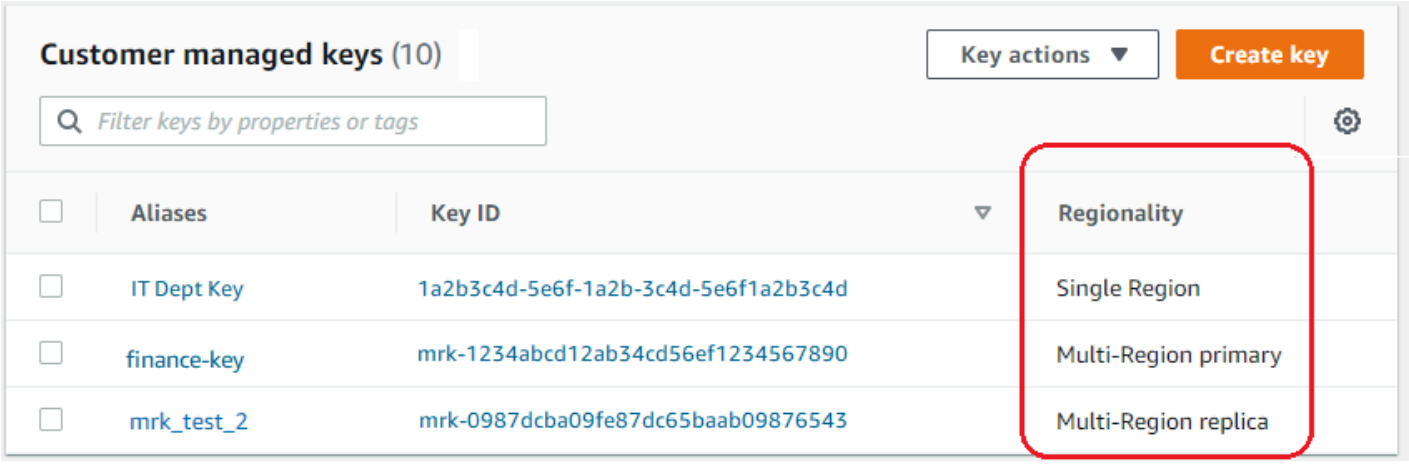
Melihat kunci multi-Wilayah di konsol tersebut

Di AWS KMS konsol, Anda dapat melihat tombol KMS di Wilayah yang dipilih. Namun, jika Anda memiliki kunci multi-Wilayah, Anda dapat melihat kunci multi-Wilayah terkait di Wilayah AWS.

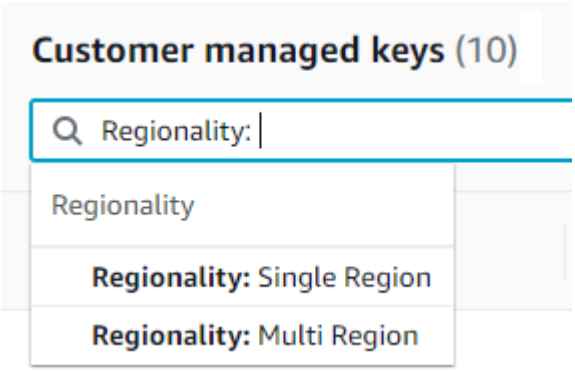
Tabel kunci terkelola Pelanggan di AWS KMS konsol hanya menampilkan kunci KMS di Wilayah yang dipilih. Anda dapat melihat kunci primer dan replika multi-Wilayah di Wilayah yang dipilih. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.

Kunci yang dikelola AWSTabel tidak memiliki fitur regionalitas karena selalu Kunci yang dikelola AWS merupakan kunci wilayah tunggal.

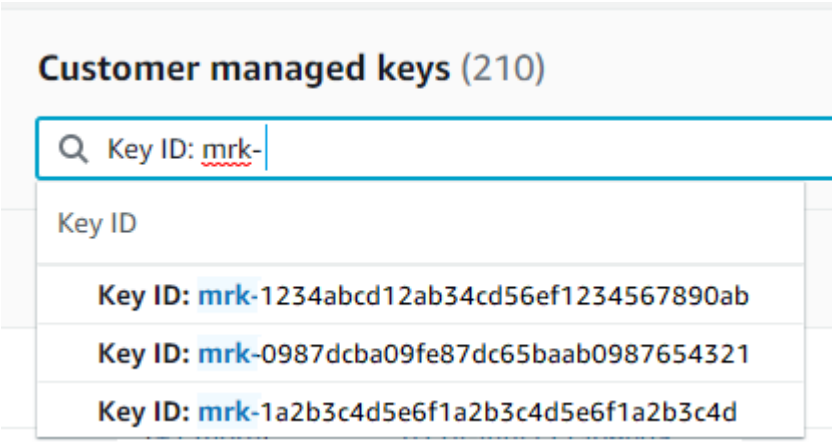
- Untuk memudahkan dalam mengidentifikasi kunci multi-Wilayah Anda, tambahkan kolom Regionalitas ke tabel kunci Anda. Untuk bantuan, lihat Menyesuaikan tabel kunci KMS Anda.



- Untuk menampilkan kunci Wilayah tunggal atau kunci multi-Wilayah saja di tabel kunci Anda, filter kunci Anda dengan properti Regionalitas dari setiap kunci. Untuk bantuan, lihat Menyortir dan memfilter kunci KMS Anda.



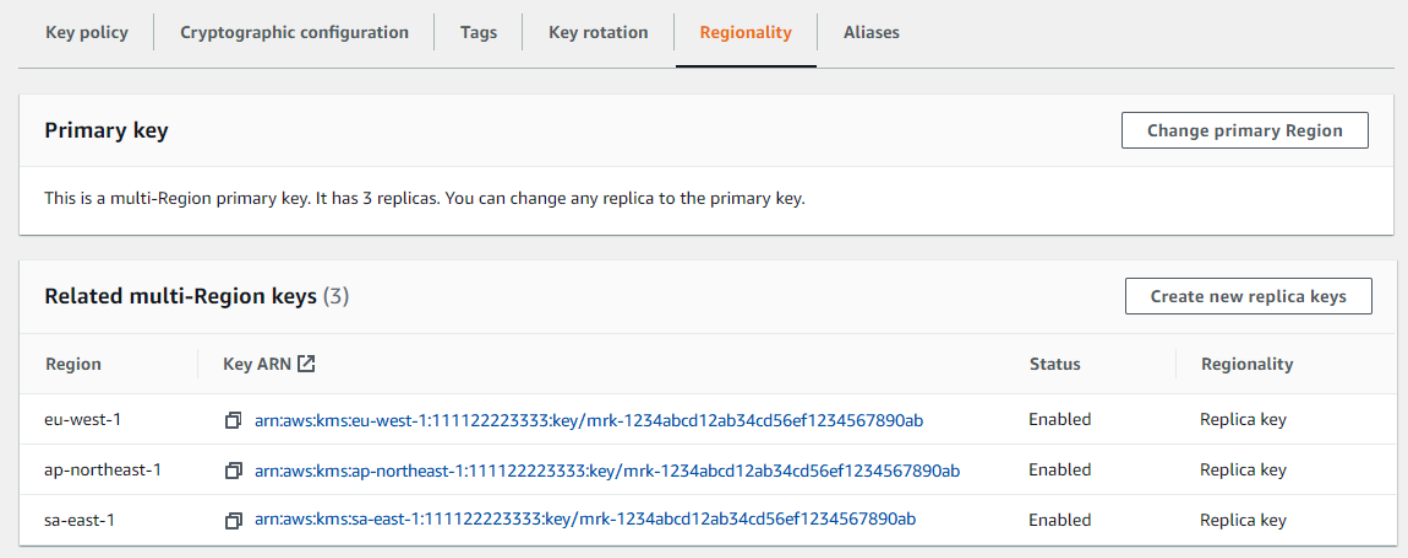
- Anda juga dapat mengurutkan dan memfilter tabel kunci terkelola Pelanggan Anda untuk awalan ID kunci mrk yang khas.



- Untuk detail tentang kunci primer multi-Wilayah atau kunci replika, buka halaman detail untuk kunci, dan pilih tab Regionalitas.

Tab Regionalitas untuk kunci primer termasuk Mengubah Wilayah primer dan Membuat tombol kunci replika baru. (Tab Regionalitas untuk kunci replika tidak memiliki tombol apa pun.) Bagian Kunci multi-Wilayah terkait mencantumkansemua kunci multi-Wilayah yang terkait dengan yang sekarang. Jika kunci saat ini adalah kunci replika, daftar ini mencakup kunci primer.

Jika Anda memilih kunci multi-Wilayah terkait dari tabel Kunci multi-Wilayah yang terkait, konsol AWS KMS tersebut berubah menjadi Wilayah dari kunci yang dipilih dan membuka halaman detail untuk kunci. Sebagai contoh, jika Anda memilih kunci replika di Wilayah `sa-east-1` dari bagian contoh Kunci Multi-wilayah yang terkait di bawah ini, konsol AWS KMS tersebut berubah menjadi Wilayah `sa-east-1` untuk menampilkan halaman detail untuk kunci replika itu. Anda dapat melakukan ini untuk melihat alias atau kebijakan kunci untuk kunci replika. Untuk mengubah Wilayah lagi, gunakan pemilih Wilayah di sudut kanan atas halaman.



Melihat kunci multi-Wilayah di API

Untuk melihat kunci Multi-region di AWS KMS API, gunakan DescribeKeyoperasi. Ini menampilkan kunci tertentu dan semua kunci multi-Wilayah terkaitnya.

Seperti konsol AWS KMS tersebut, operasi API AWS KMS bersifat Regional. Misalnya, ketika Anda memanggil ListAliasesoperasi ListKeysatau, mereka hanya mengembalikan sumber daya di Wilayah saat ini atau yang ditentukan. Namun, ketika Anda memanggil operasi `DescribeKey` pada kunci multi-Wilayah, responsnya mencakup semua kunci multi-Wilayah terkait di Wilayah AWS lain.

Misalnya, permintaan `DescribeKey` berikut mendapatkan detail tentang contoh kunci replika multi-Wilayah di Asia Pacific (Tokyo) Wilayah (`ap-northeast-1`).

```
$ aws kms describe-key \
        --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
        --region ap-northeast-1
```

Sebagian besar `KeyMetadata` dalam tanggapan tersebut menjelaskan kunci replika di Wilayah Asia Pacific (Tokyo) yang menjadi subyek permintaan. Namun, elemen `MultiRegionConfiguration` menggambarkan kunci primer di Wilayah US West (Oregon) (us-west-2) dan kunci replikanya di Wilayah AWS lain, termasuk replika di Wilayah Asia Pacific (Tokyo). `DescribeKey` mengembalikan nilai `MultiRegionConfiguration`yang sama untuk semua kunci multi-Wilayah terkait.

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

Mengelola kunci multi-Wilayah

Untuk sebagian besar tindakan, Anda mengelola kunci multi-Wilayah dengan cara yang sama seperti Anda menggunakan dan mengelola kunci Wilayah tunggal. Anda dapat mengaktifkan dan menonaktifkan kunci, mengatur dan memperbarui alias, kebijakan kunci, izin, dan tanda. Namun, manajemen kunci multi-Wilayah berbeda pada aspek berikut.

- Anda dapat perbarui Wilayah primer. Hal ini mengubah salah satu kunci replika ke kunci primer dan kunci primer saat ini ke replika.
- Anda mengelola rotasi kunci otomatis hanya pada kunci primer.
- Anda bisa mendapatkan kunci publik untuk kunci multi-Wilayah asimetris dari salah satu kunci primer atau replika terkait.

Properti Multi-region yang Anda tetapkan saat membuat kunci KMS tidak dapat diubah. Anda tidak dapat mengonversi kunci Single-region menjadi kunci Multi-region atau mengonversi kunci Multi-region menjadi kunci Single-region.

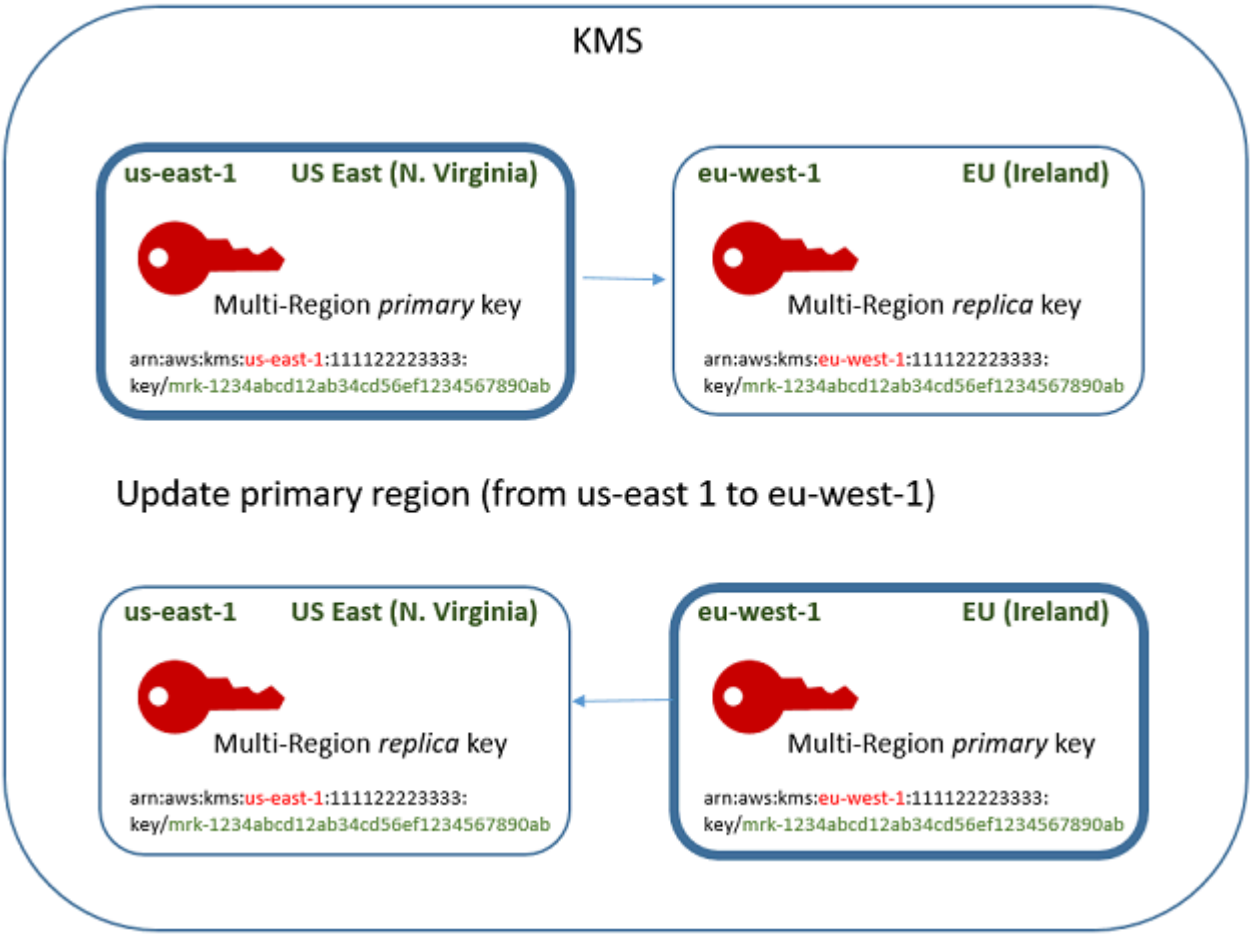
Memperbarui Wilayah primer

Setiap kumpulan kunci multi-Wilayah terkait harus memiliki kunci primer. Tapi Anda dapat mengubah kunci primer. Tindakan ini, yang dikenal sebagai memperbarui Wilayah primer, mengubah kunci primer saat ini ke kunci replika dan mengubah salah satu kunci replika terkait ke kunci primer. Anda harus melakukan ini jika Anda perlu menghapus kunci primer saat ini sambil mempertahankan kunci replika, atau untuk menemukan kunci primer di Wilayah yang sama dengan administrator kunci Anda.

Anda dapat memilih kunci replika terkait untuk menjadi kunci primer baru. Kunci primer dan kunci replika harus dalam status kunci `Enabled` saat operasi dimulai.

Bahkan setelah operasi ini selesai, proses memperbarui Wilayah primer mungkin masih berlangsung selama beberapa detik lagi. Selama waktu ini, kunci primer lama dan baru memiliki status kunci sementara Memperbarui. Sementara status kunci adalah `Updating`, Anda dapat menggunakan kunci dalam operasi kriptografi, tetapi Anda tidak dapat mereplikasi kunci primer baru atau melakukan operasi manajemen tertentu, seperti mengaktifkan atau menonaktifkan kunci ini. Operasi seperti DescribeKeymungkin menampilkan kunci primer lama dan baru sebagai replika. Status kunci `Enabled` dipulihkan ketika pembaruan selesai.

Misalkan Anda memiliki kunci primer di US East (N. Virginia) (us-east-1) dan kunci replika di Eropa (Irlandia) (eu-west-1). Anda dapat menggunakan fitur pembaruan untuk mengubah kunci primer pada US East (N. Virginia) (us-east-1) ke kunci replika dan mengubah kunci replika di Eropa (Irlandia) (eu-barat-1) ke kunci primer.



Ketika proses pembaruan selesai, kunci multi-Wilayah di Wilayah Eropa (Irlandia) (eu-west-1) adalah kunci primer multi-Wilayah dan kunci di Wilayah US East (N. Virginia) (us-east-1) adalah

kunci replikanya. Jika ada kunci replika terkait lainnya, maka mereka menjadi replika kunci primer baru. Lain kali yang AWS KMS menyinkronkan properti bersama dari kunci Multi-region, ia akan mendapatkan properti bersama dari kunci utama baru dan menyalinnya ke kunci replika, termasuk kunci primer sebelumnya.

Operasi pembaruan tidak berpengaruh pada ARN kunci dari setiap kunci multi-Wilayah. Ini juga tidak berpengaruh pada properti bersama, seperti materi kunci, atau pada properti independen, seperti kebijakan kunci. Namun, Anda mungkin ingin memperbarui kebijakan kunci dari kunci primer baru. Misalnya, Anda mungkin ingin menambahkan ReplicateKey izin kms: untuk prinsipal tepercaya ke kunci utama baru dan menghapusnya dari kunci replika baru.

Status kunci **`Updating`**

Proses memperbarui Wilayah primer membutuhkan waktu sedikit lebih lama daripada penundaan konsistensi akhirnya yang singkat yang memengaruhi sebagian besar AWS KMS operasi. Proses mungkin masih berlangsung setelah operasi `UpdatePrimaryRegion` mengembalikan atau Anda telah menyelesaikan prosedur pembaruan di konsol tersebut. Operasi seperti DescribeKeymungkin menampilkan kunci primer lama dan baru sebagai replika sampai proses selesai.

Selama proses memperbarui Wilayah primer, kunci primer lama dan kunci primer baru berada di status kunci `Updating`. Ketika proses pembaruan selesai berhasil, kedua kunci kembali ke status `Enabled`. Sementara di status `Updating`, beberapa operasi manajemen, seperti mengaktifkan dan menonaktifkan kunci, tidak tersedia. Namun, Anda dapat terus menggunakan kedua kunci dalam operasi kriptografi tanpa gangguan. Untuk informasi tentang efek dari status kunci `Updating`, lihat Status AWS KMS kunci kunci.

Memperbarui Wilayah primer (konsol)

Anda dapat memperbarui kunci utama di AWS KMS konsol. Mulai pada halaman detail kunci untuk kunci primer saat ini.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Pada panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih ID kunci atau alias dari kunci primer multi-Wilayah. Ini membuka halaman detail kunci untuk kunci utama.

Untuk mengidentifikasi kunci primer multi-Wilayah, gunakan ikon alat di sudut kanan atas untuk menambahkan kolom Regionalitas ke tabel.

5. Pilih tab Regionalitas.
6. Pada bagian Kunci primer, pilih Ubah Wilayah primer.
7. Pilih Wilayah kunci primer baru. Anda hanya dapat memilih satu Wilayah dari menu.

   Menu Ubah Wilayah primer hanya mencakup Wilayah yang memiliki kunci multi-Wilayah terkait. Anda mungkin tidak memiliki izin untuk memperbarui Wilayah primer di semua Wilayah pada menu.
8. Pilih Ubah Wilayah primer.

Memperbarui Wilayah (AWS KMS API) utama

Untuk mengubah kunci utama dalam satu set kunci Multi-wilayah terkait, gunakan UpdatePrimaryRegionoperasi.

Gunakan parameter `KeyId` untuk mengidentifikasi kunci primer saat ini. Gunakan `PrimaryRegion` parameter untuk menunjukkan Wilayah AWS kunci utama baru. Jika kunci primer belum memiliki replika di Wilayah primer baru, operasi gagal.

Contoh berikut merubah kunci primer dari kunci multi-Wilayah di Wilayah `us-west-2` untuk replikanya di Wilayah `eu-west-1`. Parameter `KeyId` mengidentifikasi kunci primer saat ini di Wilayah `us-west-2`. `PrimaryRegion`Parameter menentukan kunci Wilayah AWS utama baru,`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
```

Ketika berhasil, operasi ini tidak mengembalikan output apapun; hanya kode status HTTP. Untuk melihat efeknya, panggil DescribeKeyoperasi pada salah satu tombol Multi-region. Anda mungkin ingin menunggu sampai status kunci kembali ke `Enabled`. Sementara status kunci adalah Memperbarui, nilai untuk kunci mungkin masih dalam fluks.

Misalnya, `DescribeKey` berikut mendapat detail tentang kunci multi-Wilayah di Wilayah `eu-west-1`. Output menunjukkan bahwa kunci multi-Wilayah dalam Wilayah `eu-west-1` sekarang

adalah kunci primer. Kunci multi-Wilayah terkait (ID kunci yang sama) di Wilayah `us-west-2` sekarang adalah kunci replika.

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

Memutar kunci multi-Wilayah

Anda dapat mengaktifkan dan menonaktifkan rotasi otomatis dan melakukan rotasi sesuai permintaan dari materi kunci di tombol Multi-wilayah. Rotasi kunci adalah properti bersama dari kunci Multi-wilayah.

Anda dapat mengaktifkan dan menonaktifkan rotasi kunci otomatis hanya pada kunci primer. Anda memulai rotasi sesuai permintaan hanya pada kunci utama.

- Saat AWS KMS menyinkronkan kunci Multi-region, ia menyalin pengaturan properti rotasi kunci dari kunci utama ke semua kunci replika terkait.
- Saat AWS KMS memutar materi kunci, ia menciptakan materi kunci baru untuk kunci utama dan kemudian menyalin materi kunci baru melintasi batas Wilayah ke semua kunci replika terkait. Materi utama tidak pernah meninggalkan tidak AWS KMS terenkripsi. Langkah ini dikontrol dengan hati-hati untuk memastikan bahwa materi kunci sepenuhnya disinkronkan sebelum kunci apa pun digunakan dalam operasi kriptografi.
- AWS KMS tidak mengenkripsi data apa pun dengan materi kunci baru sampai materi kunci tersebut tersedia di kunci utama dan setiap kunci replika.
- Ketika Anda mereplikasi kunci primer yang telah diputar, kunci replika baru memiliki materi kunci saat ini dan semua versi dari materi kunci sebelumnya untuk kunci multi-Wilayah terkait.

Pola ini memastikan bahwa kunci multi-Wilayah terkait sepenuhnya dapat dioperasikan. Kunci multi-Wilayah apa pun dapat mendekripsi ciphertext yang dienkripsi oleh kunci multi-Wilayah terkait, bahkan jika ciphertext telah dienkripsi sebelum kunci dibuat.

Rotasi tombol otomatis tidak didukung pada tombol KMS asimetris atau kunci KMS dengan bahan kunci impor. Untuk informasi tentang rotasi kunci otomatis dan sesuai permintaan, lihatBerputar AWS KMS keys.

Mengunduh kunci publik

Saat Anda membuat kunci KMS asimetris Multi-region, AWS KMS buat key pair RSA atau eliptic curve (ECC) untuk kunci primer. Kemudian ia menyalin pasangan kunci tersebut ke setiap replika kunci primer. Sebagai hasilnya, Anda dapat mengunduh kunci publik dari kunci primer atau salah satu kunci replikanya. Anda akan selalu mendapatkan materi kunci yang sama.

Untuk informasi tentang mengunduh dan menggunakan kunci publik di luar AWS KMS, lihatPertimbangan khusus untuk mengunduh kunci publik. Untuk petunjuk, lihat Mengunduh kunci publik.

Mengimpor materi kunci ke kunci multi-Wilayah

Anda dapat mengimpor materi kunci Anda sendiri ke kunci KMS Multi-wilayah. Kunci multi-Wilayah yang Anda buat dengan materi kunci Anda sendiri dapat dioperasikan. Anda dapat menggunakan kunci multi-Wilayah terkait untuk mengenkripsi data dalam satu Wilayah dan mendekripsi data di Wilayah lain.

Namun, Anda harus mengelola materi kunci.

- AWS KMS tidak menyalin atau menyinkronkan materi kunci dari kunci primer dengan materi kunci yang diimpor ke kunci replika. Anda harus mengimpor materi kunci yang sama ke kunci primer dan replika terkait.
- Anda mengatur model kedaluwarsa dan tanggal kedaluwarsa untuk setiap kunci secara independen saat Anda mengimpor materi kunci. Anda dapat mengkonfigurasi model kedaluwarsa dan tanggal kedaluwarsa yang sama maupun berbeda untuk kunci multi-Wilayah terkait. Jika materi kunci mendekati tanggal kedaluwarsa, Anda harus mengimpor ulang materi kunci ke kunci multi-Wilayah yang terpengaruh.

  Status kunci dari kunci multi-Wilayah terkait, terpisah satu sama lain. Misalya, jika materi kunci dalam kunci primer kedaluwarsa, maka kunci replikanya tidak terpengaruh.

Persyaratan Wilayah untuk kunci replika yang sama berlaku untuk kunci multi-Wilayah dengan materi kunci yang diimpor. Jika Anda mengimpor materi kunci yang sama ke kunci wilayah tunggal atau kunci Multi-wilayah yang tidak terkait, kunci KMS ini tidak dapat dioperasikan.

Anda dapat membuat kunci Multi-Region dengan bahan kunci simetris, asimetris, atau HMAC yang diimpor. AWS KMStidak mendukung materi kunci yang diimpor di toko kunci khusus. Selain itu, Anda tidak dapat mengaktifkan rotasi tombol otomatis dari kunci KMS apa pun dengan bahan kunci yang diimpor.

Selain fitur Multi-region mereka, kunci Multi-region dengan material kunci impor sama dengan kunci KMS lainnya dengan bahan kunci impor. Untuk informasi rinci tentang membuat dan mengonfigurasi kunci wilayah tunggal dengan materi kunci yang diimpor, lihat. Tentang material kunci yang diimpor

Topik

Mengapa tidak semua kunci KMS dengan bahan kunci impor dapat dioperasikan?

Kunci KMS wilayah tunggal dengan bahan kunci impor tidak dapat dioperasikan, bahkan ketika mereka memiliki bahan kunci yang sama. Ketika AWS KMS menggunakan kunci KMS untuk mengenkripsi data, secara kriptografis mengikat beberapa metadata kunci ke ciphertext. Ini mengamankan ciphertext sehingga hanya kunci KMS yang data terenkripsi dapat mendekripsi data tersebut.

Kunci multi-Wilayah dirancang agar dapat dioperasikan. Selain memiliki materi kunci yang sama, mereka memiliki ID kunci yang sama dan metadata lainnya. Dengan demikian, ciphertext yang mereka hasilkan dapat didekripsi oleh kunci multi-Wilayah terkait. Akibatnya, properti kepercayaan kunci multi-Wilayah berbeda dari kunci wilayah tunggal. Tetapi bagi sebagian pelanggan, manfaat mendekripsi di beberapa Wilayah lebih besar daripada nilai keamanan ciphertext yang bergantung pada satu kunci KMS dalam satu. Wilayah AWS

Membuat kunci primer dengan materi kunci yang diimpor

Untuk membuat kunci utama dengan materi kunci impor, Anda mulai dengan membuat kunci kunci KMS tanpa bahan kunci. Saat Anda membuat kunci utama tanpa materi kunci, Anda harus menentukan spesifikasi kunci yang mencerminkan jenis materi kunci yang akan Anda impor. Kemudian, impor materi kunci Anda ke kunci utama.

Prosedur untuk membuat kunci primer multi-Wilayah tanpa matteri kunci hampir sama dengan prosedur membuat kunci Wilayah tunggal tanpa materi kunci. Satu-satunya perbedaan adalah Anda menentukan bahwa kuncinya adalah kunci Multi-wilayah.

Izin untuk membuat kunci primer Multi-wilayah dengan materi kunci impor sama dengan yang diperlukan untuk membuat kunci primer Multi-wilayah dengan materi AWS KMS kunci, termasuk CreateServiceLinkedRole izin kms: CreateKey dan iam: dalam kebijakan IAM. Anda dapat menggunakan kunci KeyOrigin kondisi kms: MultiRegionKeyType dan kms: untuk mengizinkan atau menolak izin untuk membuat kunci utama Multi-wilayah dengan materi kunci yang diimpor.

Saat membuat kunci utama dengan materi kunci yang diimpor di AWS KMS konsol, gunakan pengaturan di bagian Opsi lanjutan. Anda tidak dapat mengubah properti ini setelah kunci KMS dibuat.

- Atur asal bahan utama ke Eksternal (Impor bahan kunci).
- Atur Replikasi multi-Wilayah, pilih Izinkan kunci ini untuk direplikasi ke Wilayah lain.

Saat menggunakan CreateKeyoperasi untuk membuat kunci utama dengan bahan kunci yang diimpor, gunakan `Origin` dan `MultiRegion` parameter dan tentukan `KeySpec` dan`KeyUsage`. Contoh berikut membuat kunci `EXTERNAL` KMS yang dapat mengimpor materi `ECC_NIST_P384` kunci.

```
$ aws kms create-key --origin EXTERNAL --key-spec ECC_NIST_P384 --key-usage SIGN_VERIFY
 --multi-region
```

Hasilnya adalah kunci primer multi-Wilayah tanpa materi kunci dan status kunci dari `PendingImport`.

Untuk mengaktifkan kunci KMS ini, Anda harus mengunduh kunci publik dan token impor, menggunakan kunci publik untuk mengenkripsi materi kunci Anda, dan kemudian mengimpor materi kunci Anda. Untuk mengetahui petunjuknya, lihat Mengimpor bahan kunci untuk AWS KMS kunci.

Membuat kunci replika dengan materi kunci yang diimpor

Anda dapat membuat kunci replika multi-Wilayah di konsol AWS KMS tersebut atau menggunakan operasi API AWS KMS. Untuk mereplikasi kunci primer multi-wilayah dengan materi kunci yang diimpor, Anda menggunakan prosedur yang sama yang Anda gunakan untuk membuat kunci replika dengan materi kunci AWS KMS. Namun, hasilnya berbeda. Proses replikasi tidak mengembalikan kunci replika dengan materi kunci yang sama dengan kunci primer melainkan mengembalikan kunci replika tanpa materi kunci dan status kunci dari `PendingImport`. Untuk mengaktifkan kunci replika, Anda harus mengimpor bahan kunci yang sama ke kunci replika yang Anda impor ke kunci utamanya.

Meskipun tidak mereplikasi materi kunci, AWS KMS membuat kunci replika dengan ID kunci, spesifikasi kunci, penggunaan kunci, dan asal materi kunci yang sama dengan kunci primer. Hal ini juga memastikan bahwa materi kunci yang Anda impor ke kunci replika identik dengan materi kunci yang Anda impor ke kunci primer.

Untuk membuat kunci replika dengan materi kunci yang diimpor:

1. Buat kunci primer multi-Wilayah dengan materi kunci yang diimpor.
2. Lakukan salah satu dari berikut ini.

Pada konsol AWS KMS tersebut, pilih kunci primer multi-Wilayah dengan materi kunci yang diimpor. Kemudian, pada tab Regionalitas-nya, pilih Buat kunci replika baru. Untuk mengetahui petunjuknya, lihat Membuat kunci replika (konsol).

Atau gunakan ReplicateKeyoperasi. Untuk parameter `KeyId`, masukkan ID kunci atau ARN kunci dari kunci primer multi-Wilayah dengan materi kunci yang diimpor. Untuk mengetahui petunjuknya, lihat Membuat kunci replika (API AWS KMS).

3. Untuk setiap kunci replika baru, ikuti langkah-langkah untuk mengunduh kunci publik dan token impor. Gunakan kunci publik untuk mengenkripsi materi kunci primer, dan kemudian impor materi kunci primer di kunci replika. Anda memerlukan kunci publik dan token impor yang berbeda untuk setiap kunci replika.

   Jika materi kunci yang Anda coba untuk impor ke kunci replika bukan materi kunci yang sama dengan kunci primer, operasi akan gagal. AWS KMS tidak mengharuskan model kedaluwarsa dan tanggal kedaluwarsa dikoordinasikan, tetapi Anda mungkin menetapkan aturan bisnis untuk kunci multi-Wilayah Anda. Untuk mengetahui petunjuknya, lihat Mengimpor bahan kunci untuk AWS KMS kunci.

Izin untuk mereplikasi kunci dengan materi kunci yang diimpor

Untuk membuat kunci replika dengan materi kunci yang diimpor, Anda harus memiliki izin berikut.

Pada Wilayah kunci primer:

- kms: ReplicateKey pada kunci utama (di Region kunci utama). Sertakan izin ini dalam kebijakan kunci utama atau dalam kebijakan IAM.

Pada Wilayah kunci replika:

- kms: CreateKey dalam kebijakan IAM.
- km: GetParametersForImport. Anda dapat menyertakan izin ini dalam kebijakan kunci dari replika kunci atau kebijakan IAM.
- km: ImportKeyMaterial. Anda dapat menyertakan izin ini dalam kebijakan kunci dari replika kunci atau kebijakan IAM.
- kms: TagResource diperlukan untuk menetapkan tag saat mereplikasi. Sertakan izin ini dalam kebijakan IAM di Wilayah replika.

- kms: CreateAlias diperlukan untuk mereplikasi kunci di konsol. AWS KMS Lihat perinciannya di Mengontrol akses ke alias.

Menghapus kunci multi-Wilayah

Ketika Anda tidak lagi menggunakan kunci primer multi-Wilayah atau kunci replika, Anda dapat menjadwalkan penghapusan.

Meskipun menghapus kunci KMS harus selalu dilakukan dengan hati-hati, menghapus replika kunci Multi-wilayah kurang berisiko, asalkan kunci utama masih ada di. AWS KMS Jika Anda menghapus kunci replika dari Wilayah-nya, tetapi menemukan ciphertext yang dienkripsi di bawah kunci terhapus, Anda dapat mendekripsi ciphertext tersebut dengan kunci multi-Wilayah terkait apa pun. Anda juga dapat membuat kunci replika dengan mereplikasi kunci primer lagi ke replika kunci Wilayah.

Namun, menghapus kunci utama dan semua kunci replika adalah operasi yang sangat berbahaya — setara dengan menghapus kunci wilayah Tunggal.

> **Warning**
>
> Menghapus kunci KMS bersifat merusak dan berpotensi berbahaya. Anda harus melanjutkan hanya ketika Anda yakin bahwa Anda tidak perlu menggunakan kunci KMS lagi dan tidak perlu menggunakannya di masa depan. Jika Anda tidak yakin, Anda harus menonaktifkan kunci KMS alih-alih menghapusnya.

Untuk menghapus kunci primer, Anda harus terlebih dahulu menghapus semua kunci replika. Jika Anda harus menghapus kunci utama dari Wilayah tertentu tanpa menghapus kunci replika, ubah kunci utama menjadi kunci replika dengan memperbarui Wilayah utama.

Sebelum Anda menjadwalkan penghapusan kunci KMS apa pun, tinjau peringatan dalam Menghapus AWS KMS keys topik, dan topik yang menjelaskan cara menentukan penggunaan kunci KMS di masa lalu dan cara mengatur CloudWatch alarm yang mengingatkan Anda untuk menggunakan kunci KMS selama masa tunggu. Sebelum menghapus kunci utama kunci Multi-region asimetris, tinjau topik Menghapus kunci asimetris.

Topik

- Izin untuk menghapus kunci multi-Wilayah
- Cara menghapus kunci replika

- Cara menghapus kunci primer

Izin untuk menghapus kunci multi-Wilayah

Untuk menjadwalkan penghapusan kunci multi-Wilayah, Anda hanya memerlukan izin berikut.

- kms: ScheduleKeyDeletion — untuk menjadwalkan penghapusan kunci Multi-wilayah dan mengatur masa tunggunya.

Kami juga sangat menyarankan bahwa Anda harus memiliki izin terkait berikut.

- kms: CancelKeyDeletion — untuk membatalkan penghapusan terjadwal dari kunci Multi-wilayah.
- kms: DescribeKey — untuk melihat status kunci dari kunci Multi-wilayah dan daftar kunci Multi-wilayah terkait.
- kms: DisableKey — untuk memberi Anda opsi untuk menonaktifkan kunci Multi-wilayah alih-alih menghapusnya.
- kms: EnableKey — untuk mengembalikan fungsionalitas kunci Multi-wilayah setelah membatalkan penghapusannya.

Anda mungkin juga menyertakan izin untuk mereplikasi kunci primer serta mengubah kunci primer.

- km: ReplicateKey
- km: UpdateReplicaRegion

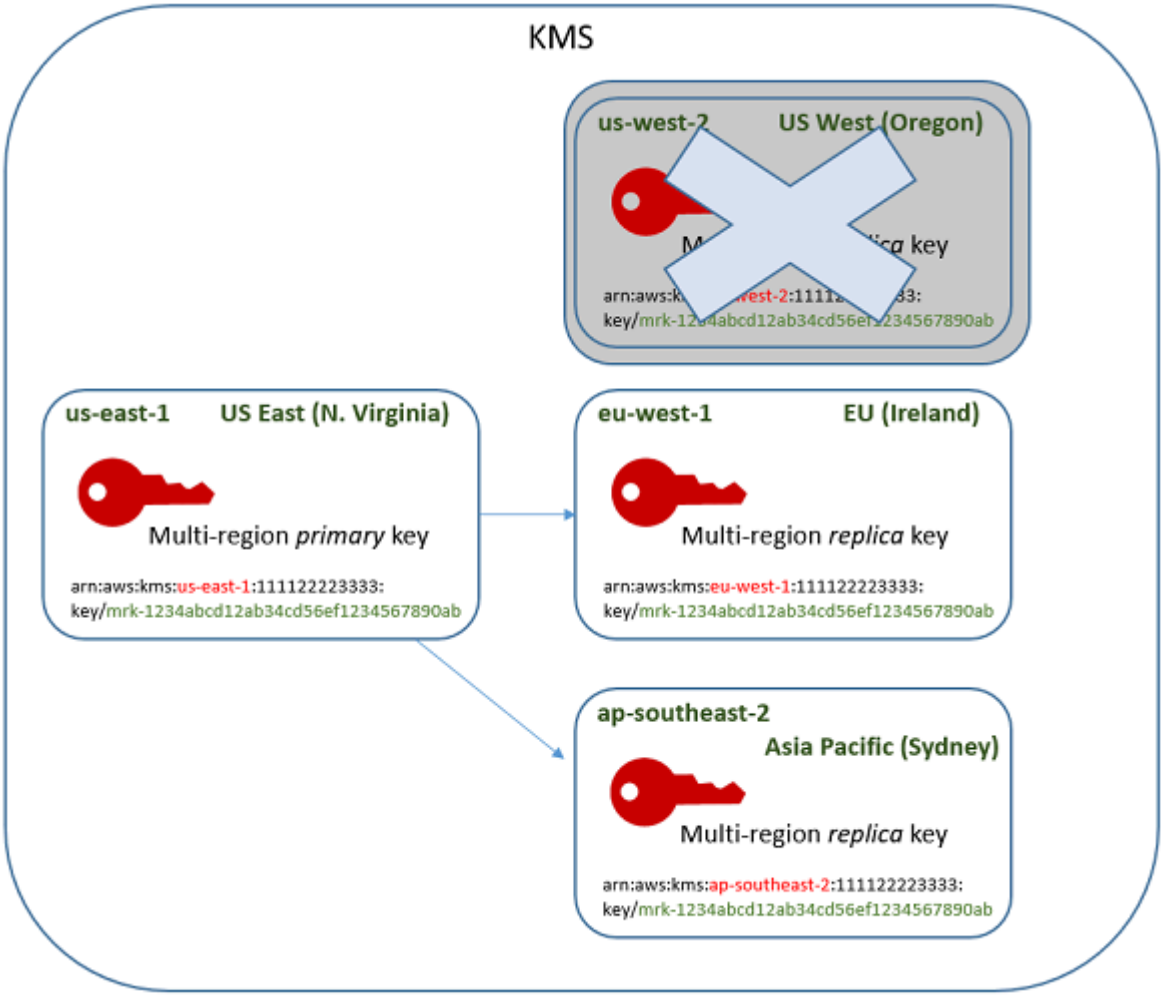
Anda dapat menyertakan izin ini dalam kebijakan IAM, tetapi merupakan praktik terbaik untuk menempatkannya dalam kebijakan utama yang hanya berlaku untuk kunci KMS yang perlu Anda kelola.

Cara menghapus kunci replika

Anda dapat menggunakan konsol AWS KMS atau API AWS KMS untuk menghapus kunci replika. Anda dapat menghapus kunci replika kapan saja. Itu tidak tergantung pada status kunci dari kunci KMS lainnya.

Jika Anda salah menghapus kunci replika, Anda dapat membuatnya kembali dengan mereplikasi kunci utama yang sama di Wilayah yang sama. Kunci replika baru yang Anda buat akan memiliki properti bersama yang sama dengan kunci replika asli.

Prosedur untuk menghapus kunci replika multi-Wilayah sama dengan menghapus kunci Wilayah tunggal.



1. Jadwal penghapusan kunci replika. Pilih masa tunggu 7-30 hari. Masa tunggu default adalah 30 hari.
2. Selama masa tunggu, status kunci dari kunci replika berubah menjadi `Pending deletion` (`PendingDeletion`) dan Anda tidak dapat menggunakannya dalam operasi kriptografi.
3. Anda dapat membatalkan jadwal penghapusan kunci replika pada setiap titik di masa tunggu. Status kunci berubah menjadi`Disabled`, tetapi Anda dapat mengaktifkan kembali kunci KMS.
4. Saat masa tunggu kedaluwarsa, AWS KMS menghapus kunci replika.

Anda dapat melihat catatan tindakan Anda di AWS CloudTrail log Anda. AWS KMSmencatat operasi yang menjadwalkan penghapusan kunci KMS dan tindakan yang menghapus kunci KMS.

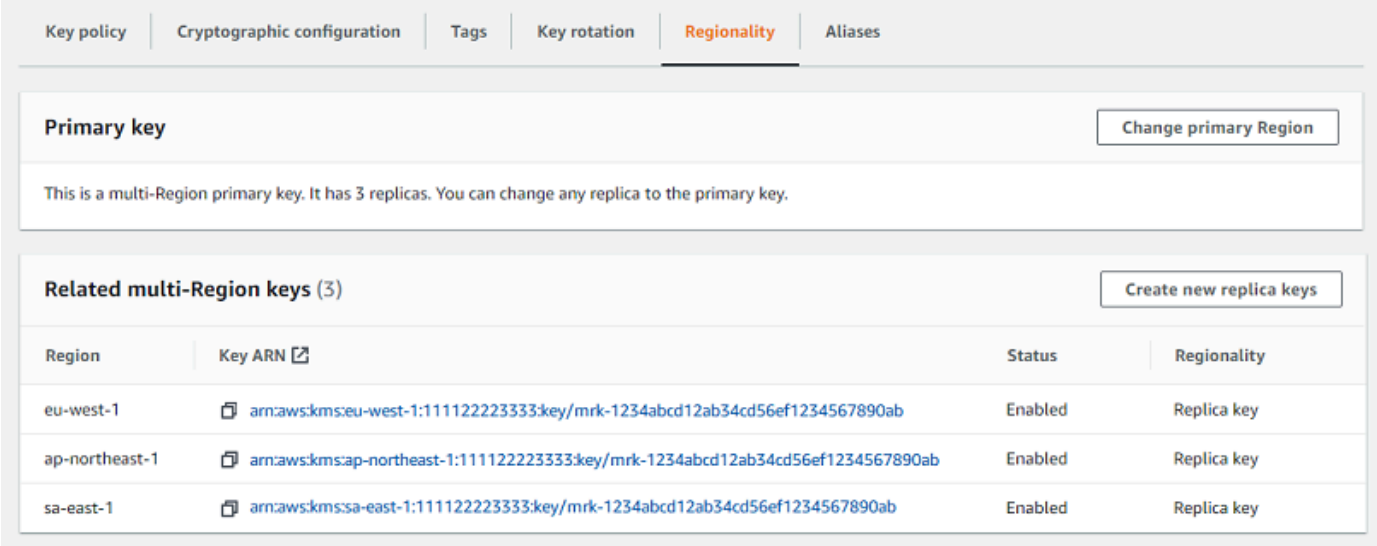
Menghapus kunci replika (konsol)

Untuk menjadwalkan penghapusan kunci replika multi-Wilayah, gunakan prosedur yang sama yang Anda gunakan untuk menjadwalkan penghapusan kunci Wilayah tunggal.

Karena kunci replika terkait bereda dengan Wilayah AWS, Anda tidak dapat menjadwalkan penghapusan lebih dari satu kunci replika pada satu waktu. Untuk menghapus semua kunci replika terkait, gunakan pola seperti berikut ini.

Untuk menjadwalkan penghapusan semua kunci replika terkait

1. Masuk ke AWS Management Console lalu buka konsol AWS Key Management Service (AWS KMS) tersebut di https://console.aws.amazon.com/kms.
2. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
3. Gunakan pemilih Wilayah di sudut kanan atas untuk memilih Wilayah dari kunci primer multi-Wilayah..
4. Pilih alias atau ID kunci dari kunci primer.
5. Pilih tab Regionalitas.



6. Pada bagian Kunci multi-Wilayah terkait, pilih ARN kunci dari kunci replika.

   Tindakan ini membuka halaman detail kunci dari replika kunci di tab peramban baru. Konsol tersebut diatur ke Wilayah kunci replika.
7. Pilih menu Tindakan kunci, pilih Menjadwalkan penghapusan kunci.

Tindakan ini memulai proses penjadwalan penghapusan kunci. Menyelesaikan proses jadwal penghapusan kunci. UNtuk detailnya, lihat Menjadwalkan dan membatalkan penghapusan kunci (konsol).

8. Kembali ke tab peramban yang menampilkan tab Regionalitas kunci primer. (Anda mungkin perlu me-refresh halaman untuk melihat status terbaru kunci replika.) Pilih ARN kunci dari kunci replika lain dan ulangi proses penjadwalan penghapusan kunci replika.

Menghapus kunci replika (API AWS KMS)

Untuk menjadwalkan penghapusan kunci replika Multi-wilayah, gunakan operasi. ScheduleKeyDeletion Untuk menentukan kunci KMS, gunakan ID kunci atau kunci ARN. Saat bekerja dengan kunci Multi-region, Anda dapat mengurangi kejadian kesalahan dengan menggunakan kunci ARN dengan nilai Region eksplisit.

Misalnya, perintah ini menghapus kunci replika dari Wilayah us-west-2 (US West (Oregon)). Karena perintah tidak menentukan masa tunggu, masa tunggu diatur ke default 30 hari.

```
$ aws kms schedule-key-deletion \
    --region us-west-2 \
    --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Ketika perintah berhasil, ia mengembalikan ARN kunci (`KeyId`), masa tunggu (`PendingWindowInDays`), tanggal penghapusan (`DeletionDate`), dan status kunci saat ini (`KeyState`), yang diharapkan ke `PendingDeletion`.

Saat menghapus kunci replika multi-Wilayah, pastikan untuk memverifikasi bahwa ID kunci dan nilai Wilayah di ARN kunci seperti yang Anda harapkan.

```
{
    "KeyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "DeletionDate": 1599523200.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 30
}
```

Untuk menghapus semua replika kunci primer multi-Wilayah secara terprogram, buat daftar wilayah yang berisi kunci replika. Kemudian, untuk setiap Wilayah dalam daftar, panggil operasi `ScheduleKeyDeletion`, seperti yang ditunjukkan di atas.

Tidak seperti kunci Single-region yang dihapus secara permanen, Anda dapat memulihkan kunci replika dengan mereplikasi kunci utama ke Wilayah tempat kunci replika yang dihapus berada.

Untuk memeriksa status kunci replika dan melihat kunci utama dan kunci replika dari kunci Multi-wilayah, gunakan operasi. DescribeKey

Cara menghapus kunci primer

Anda dapat menjadwalkan penghapusan kunci primer multi-Wilayah kapan saja. Namun, tidak AWS KMS akan menghapus kunci utama Multi-wilayah yang memiliki kunci replika, bahkan jika mereka dijadwalkan untuk dihapus.

Untuk menghapus kunci utama, Anda harus menjadwalkan penghapusan semua kunci replika, dan kemudian menunggu kunci replika dihapus. Masa tunggu yang diperlukan untuk menghapus kunci primer dimulai ketika kunci replika terakhirnya dihapus. Jika Anda harus menghapus kunci utama dari Wilayah tertentu tanpa menghapus kunci replika, ubah kunci utama menjadi kunci replika dengan memperbarui Wilayah utama.

Jika kunci utama tidak memiliki kunci replika, prosesnya identik dengan menghapus kunci replika atau menghapus kunci KMS regional apa pun.

Sementara kunci primer dijadwalkan untuk penghapusan, Anda tidak dapat menggunakannya dalam operasi kriptografi dan Anda tidak dapat mereplikasinya. Namun, kecuali mereka juga dijadwalkan untuk penghapusan, maka kunci replikanya tidak terpengaruh.

Anda dapat menggunakan AWS KMS konsol atau AWS KMS API untuk menjadwalkan penghapusan kunci primer dan replika. Anda dapat menjadwalkan penghapusan kunci utama sebelum, sesudah, atau pada saat yang sama Anda menjadwalkan penghapusan kunci replika. Proses ini mungkin terlihat seperti berikut ini.

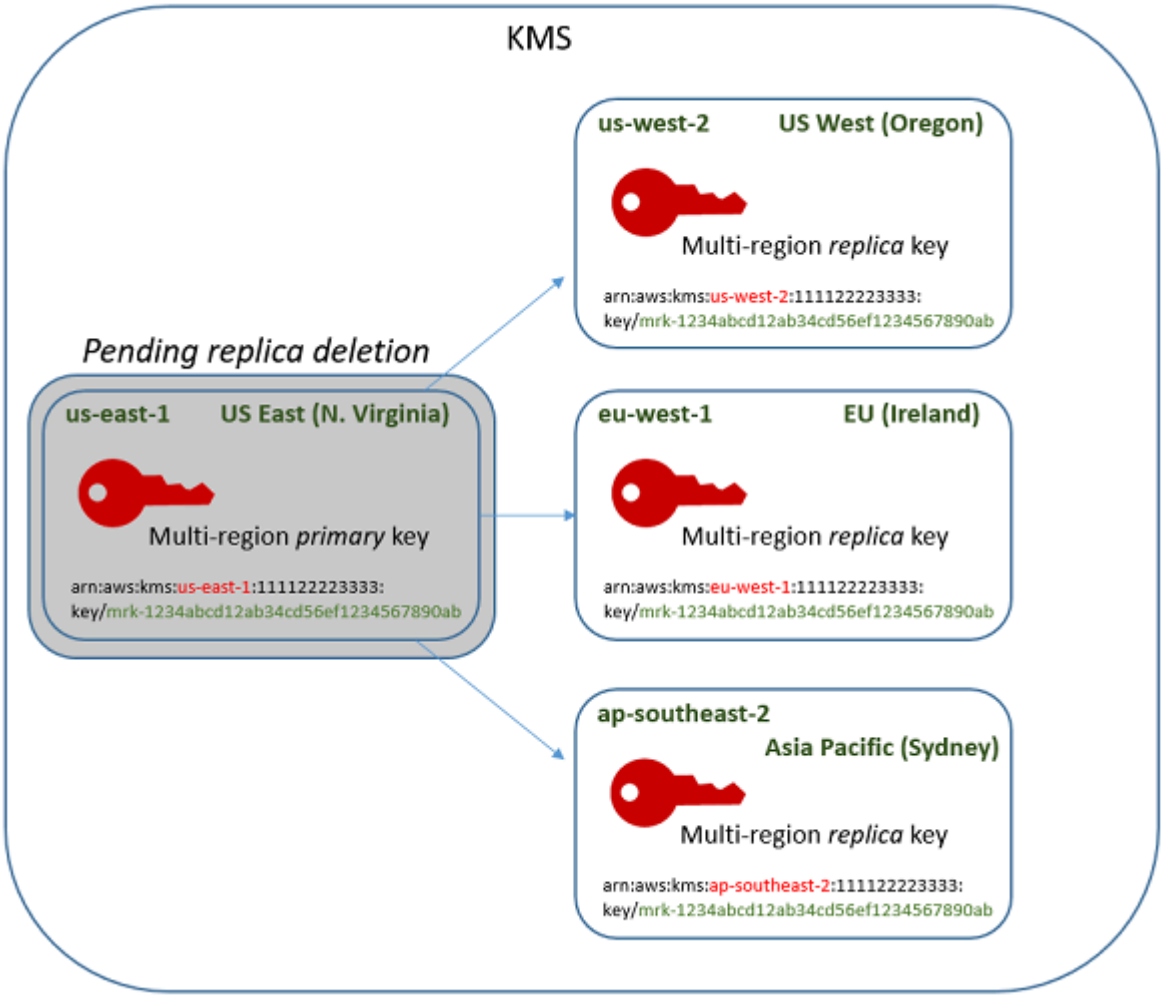
1. Menjadwalkan penghapusan kunci primer. Pilih masa tunggu 7-30 hari. Masa tunggu default adalah 30 hari. Namun, masa tunggu untuk kunci utama tidak dimulai sampai semua kunci replika dihapus.

   Jika ada kunci replika yang masih ada, status kunci dari kunci primer berubah menjadi `Pending replica deletion` (`PendingReplicaDeletion`). Jika tidak, status berubah menjadi

`Pending deletion` (`PendingDeletion`). Dalam kedua kasus, Anda tidak dapat menggunakan kunci primer dalam operasi kriptografi dan Anda tidak dapat mereplikasi itu.

Menjadwalkan penghapusan kunci primer tidak memengaruhi kunci replika. Status kunci mereka tetap diaktifkan dan Anda dapat menggunakannya dalam operasi kriptografi. Jika kunci replika tidak dihapus, `Pending replica deletion` status kunci utama dapat bertahan tanpa batas waktu.

```
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. Jadwal penghapusan setiap kunci replika. Pilih masa tunggu 7-30 hari. Masa tunggu default adalah 30 hari. Anda dapat menghapus beberapa kunci replika secara bersamaan. Masa tunggu mereka berjalan bersamaan. Selama masa tunggu, status kunci dari kunci replika berubah menjadi `Pending deletion` (`PendingDeletion`) dan Anda tidak dapat menggunakan kunci KMS ini dalam operasi kriptografi.

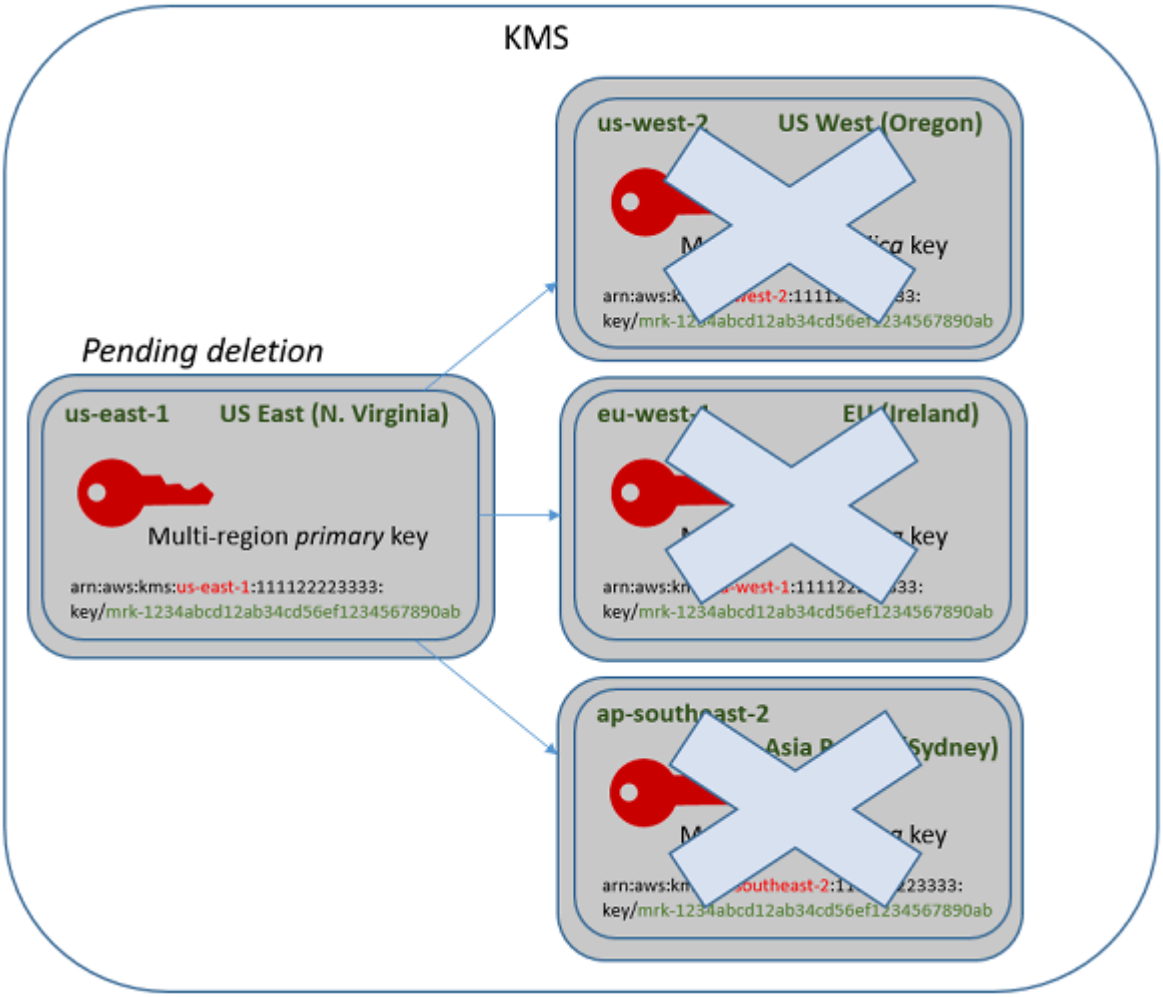
   Misalnya, jika Anda memiliki tiga kunci replika, Anda dapat menjadwalkan penghapusan ketiganya secara bersamaan. Mereka dapat memiliki periode tunggu yang sama atau berbeda. Perhatikan bahwa masa tunggu pada kunci primer belum dimulai. Status kuncinya adalah `PendingReplicaDeletion` karena ia memiliki kunci replika yang ada.

   ```
   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. Anda dapat membatalkan jadwal penghapusan kunci primer atau kunci replika sampai dihapus. Status kunci berubah menjadi`Disabled`, tetapi Anda dapat mengaktifkan kembali kunci KMS.

4. Saat masa tunggu kunci replika terakhir kedaluwarsa, AWS KMS menghapus kunci replika terakhir. Status kunci dari kunci primer berubah dari `Pending replica deletion` (`PendingReplicaDeletion`) ke `Pending deletion` (`PendingDeletion`) dan 7-30 hari masa tunggu untuk kunci primer dimulai.

   ```
   KMS key:                    Key state:
   Primary key (us-east-1)       Pending deletion (waiting period 30 days)
   ```



5. Saat masa tunggunya kedaluwarsa, AWS KMS akan menghapus kunci primer.

Waktu minimum untuk menghapus kunci primer dengan replika adalah 14 hari.

Jika Anda menjadwalkan penghapusan kunci utama dan semua kunci replika dengan masa tunggu 7 hari, kunci replika akan dihapus setelah 7 hari. Kunci primer dihapus pada hari ke 14.

- Hari 1: Jadwalkan penghapusan kunci primer dan replika dengan masa tunggu minimum 7 hari. Masa tunggu penghapusan 7 hari untuk kunci replika dimulai. Masa tunggu penghapusan untuk kunci primer belum dimulai.
- Hari 7: Masa tunggu penghapusan untuk kunci replika berakhir. AWS KMSmenghapus semua kunci replika. Ketika kunci replika terakhir dihapus, periode tunggu penghapusan 7 hari untuk kunci utama dimulai.

- Hari 14: Periode tunggu penghapusan untuk kunci primer berakhir. AWS KMS akan menghapus kunci primer.

Anda dapat melihat catatan tindakan Anda di AWS CloudTrail log Anda. AWS KMSmencatat operasi yang menjadwalkan penghapusan setiap kunci KMS dan tindakan yang menghapus kunci KMS.

Menghapus kunci primer (konsol)

Untuk menghapus kunci primer multi-Wilayah, gunakan prosedur berikut.

Untuk menjadwalkan penghapusan kunci

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih kotak centang di samping kunci primer yang ingin Anda hapus. Anda juga dapat memilih satu atau beberapa tombol KMS, termasuk replika kunci utama ini.
5. Pilih Tindakan kunci, Jadwalkan penghapusan kunci.
6. Baca dan pertimbangkan peringatan, serta informasi tentang membatalkan penghapusan selama masa tunggu. Jika Anda memutuskan untuk membatalkan penghapusan, pilih Batalkan.
7. Untuk Masa tunggu (dalam hari), masukkan beberapa hari antara 7 dan 30. Jika Anda memilih beberapa tombol KMS, masa tunggu yang Anda pilih berlaku untuk semua kunci KMS yang dipilih. Masa tunggu untuk kunci replika berjalan secara bersamaan, tetapi masa tunggu untuk kunci utama tidak dimulai sampai AWS KMS menghapus kunci replika terakhir.
8. Pilih kotak centang di samping Konfirmasi bahwa Anda ingin menjadwalkan kunci ini untuk dihapus dalam ***<number of days>*** hari.
9. Pilih Jadwalkan penghapusan.

Untuk memeriksa status penghapusan kunci KMS Anda, pada halaman detail untuk kunci utama, lihat bagian Konfigurasi umum. Status kunci muncul di bidang Status. Ketika status kunci dari kunci primer berubah menjadi `Pending deletion` Tanggal penghapusan terjadwal ditampilkan.

Anda juga dapat memeriksa status kunci (Status) dari semua kunci primer dan replika pada tab Regionalitas halaman detail untuk setiap kunci Multi-wilayah. Untuk detailnya, lihat Melihat kunci multi-Wilayah.

Menghapus kunci primer (API AWS KMS)

Untuk menghapus kunci replika Multi-wilayah, gunakan operasi. ScheduleKeyDeletion Untuk menentukan kunci KMS, gunakan ID kunci atau kunci ARN. Saat bekerja dengan kunci Multi-region, Anda dapat mengurangi kejadian kesalahan dengan menggunakan kunci ARN dengan nilai Region eksplisit.

Misalnya, perintah ini menghapus kunci primer dari Wilayah us-east-1 (US East (N. Virginia)). Karena perintah tidak menentukan masa tunggu, masa tunggu diatur ke default 30 hari.

```
$ aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Ketika perintah berhasil, ia mengembalikan ARN kunci, status kunci yang dihasilkan, dan masa tunggu (`PendingWindowInDays`).

Jika kunci primer tidak memiliki replika, status kunci dari kunci primer adalah `PendingDeletion` dan output termasuk bidang `DeletionDate`. Jika ada kunci replika yang tersisa, status kunci dari kunci utama `DeletionDate` adalah `PendingReplicaDeletion` dan dihilangkan karena tidak pasti. Bahkan jika kunci replika juga dijadwalkan untuk dihapus, Anda dapat membatalkan penghapusan terjadwal.

Saat menghapus kunci primer multi-Wilayah, pastikan untuk memverifikasi bahwa ID kunci dan nilai Wilayah di ARN kunci sudah sesuai.

```
{
    "KeyId": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyState": "PendingReplicaDeletion",
    "PendingWindowInDays": 30
}
```

Untuk memeriksa status penghapusan kunci KMS Anda, gunakan DescribeKeyoperasi pada kunci utama atau kunci replika yang tersisa. Jam periode tunggu untuk kunci utama tidak dimulai sampai replika terakhir dihapus dan status kunci berubah menjadi`PendingDeletion`.

Untuk menghitung tanggal penghapusan diharapkan kunci primer, loop melalui ARN kunci replika dalam respon, jalankan `DescribeKey` pada masing-masing, dapatkan nilai `DeletionDate` terbaru,

dan kemudian tambahkan nilai `PendingDeletionWindowInDays` untuk kunci primer. Periode tunggu untuk kunci replika berjalan secara bersamaan.

Dalam contoh berikut, kunci KMS adalah kunci utama Multi-region dengan kunci replika yang ada. Karena status kunci adalah `PendingReplicaDeletion`, respon meliputi masa tunggu (`PendingWindowInDays`), tetapi bukan `DeletionDate`. Tanggal penghapusan sebenarnya dari kunci primer tergantung pada kapan kunci replika dihapus.

```
$ 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
    }
}
```

Ketika semua replika dihapus, keluaran `DescribeKey` menunjukkan kunci primer yang tersisa dengan status kunci `PendingDeletion`. Sementara status kunci adalah `PendingDeletion`, bidang `DeletionDate` akan ditampilkan, bukan bidang `PendingWindowInDays`.

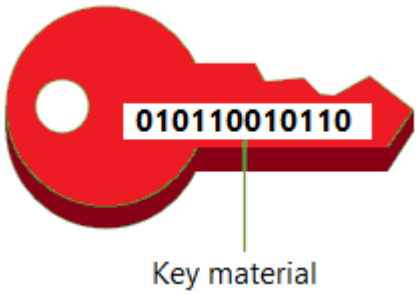
```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "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": []
        }
    }
}
```

Mengimpor bahan kunci untuk AWS KMS kunci

Anda dapat membuat AWS KMS keys(kunci KMS) dengan materi kunci yang Anda berikan.

Kunci KMS adalah representasi logis dari kunci enkripsi. Metadata untuk kunci KMS mencakup ID bahan kunci yang digunakan untuk mengenkripsi dan mendekripsi data. Saat Anda membuat kunci KMS, secara default, AWS KMS menghasilkan materi kunci untuk kunci KMS itu. Tetapi Anda dapat membuat kunci KMS tanpa materi kunci dan kemudian mengimpor materi kunci Anda sendiri ke dalam kunci KMS itu, fitur yang sering dikenal sebagai “bawa kunci Anda sendiri” (BYOK).



> **Note**
>
> AWS KMS tidak mendukung dekripsi AWS KMS ciphertext apa pun di luar AWS KMS, bahkan jika ciphertext dienkripsi di bawah kunci KMS dengan materi kunci yang diimpor. AWS KMS tidak mempublikasikan format ciphertext yang dibutuhkan tugas ini, dan formatnya mungkin berubah tanpa pemberitahuan.

Materi kunci yang diimpor didukung pada semua jenis kunci KMS kecuali untuk kunci KMS di toko kunci khusus.

Saat Anda menggunakan materi kunci impor, Anda tetap bertanggung jawab atas materi kunci sambil mengizinkan AWS KMS untuk menggunakan salinannya. Anda mungkin memilih untuk melakukan hal ini karena salah satu atau beberapa dari alasan berikut:

- Untuk membuktikan materi utama dihasilkan menggunakan sumber entropi yang memenuhi kebutuhan Anda.
- Untuk menggunakan materi utama dari infrastruktur Anda sendiri dengan AWS layanan, dan menggunakannya AWS KMS untuk mengelola siklus hidup materi utama tersebut di dalamnya. AWS
- Untuk menggunakan kunci yang sudah ada dan sudah mapan AWS KMS, seperti kunci untuk penandatanganan kode, penandatanganan sertifikat PKI, dan aplikasi yang disematkan dengan sertifikat
- Untuk mengatur waktu kedaluwarsa untuk materi kunci AWS dan menghapusnya secara manual, tetapi juga membuatnya tersedia lagi di masa mendatang. Sebaliknya, penjadwalan penghapusan kunci memerlukan masa tunggu 7 hingga 30 hari, setelah itu Anda tidak dapat memulihkan kunci KMS yang dihapus.
- Untuk memiliki salinan asli dari bahan utama, dan menyimpannya di luar AWS untuk daya tahan tambahan dan pemulihan bencana selama siklus hidup lengkap bahan utama.
- Untuk kunci asimetris dan kunci HMAC, mengimpor membuat kunci yang kompatibel dan dapat dioperasikan yang beroperasi di dalam dan di luar. AWS

Anda dapat mengaudit dan memantau penggunaan dan pengelolaan kunci KMS dengan bahan kunci impor. AWS KMS merekam peristiwa di AWS CloudTrail log Anda saat Anda membuat kunci KMS, mengunduh kunci publik pembungkus dan token impor, dan mengimpor materi kunci. AWS KMS juga mencatat peristiwa saat Anda menghapus materi kunci yang diimpor secara manual atau saat AWS KMS menghapus materi kunci yang kedaluwarsa.

Untuk informasi tentang perbedaan penting antara kunci KMS dengan bahan kunci impor dan kunci dengan bahan utama yang dihasilkan oleh AWS KMS, lihatTentang material kunci yang diimpor.

Kunci KMS yang didukung

AWS KMS mendukung bahan kunci yang diimpor untuk jenis kunci KMS berikut. Anda tidak dapat mengimpor materi kunci ke kunci KMS di toko kunci khusus.

- Kunci KMS enkripsi simetris
- Kunci KMS RSA asimetris (untuk enkripsi atau penandatanganan, tetapi tidak keduanya)
- Kunci KMS kurva elips asimetris (ECC) (hanya penandatanganan)
- Kunci SM2 KMS asimetris - Wilayah China saja (untuk enkripsi atau penandatanganan, tetapi tidak keduanya)
- Kunci HMAC KMS
- Tombol Multi-Region dari semua jenis yang didukung.

Daerah

Materi kunci yang diimpor didukung dalam semua Wilayah AWS yang AWS KMS mendukung.

Di Wilayah China, persyaratan material utama untuk kunci KMS enkripsi simetris berbeda dari Wilayah lain. Lihat perinciannya di Mengimpor material kunci langkah 3: Enkripsi material kunci.

Topik

- Berencana untuk mengimpor bahan utama
- Mengelola bahan kunci yang diimpor
- Mengimpor bahan kunci langkah 1: Buat materi AWS KMS key tanpa kunci
- Mengimpor materi kunci langkah 2: Unduh kunci publik pembungkus dan token impor
- Mengimpor material kunci langkah 3: Enkripsi material kunci
- Mengimpor material kunci langkah 4: Impor material kunci

Berencana untuk mengimpor bahan utama

Materi kunci yang diimpor memungkinkan Anda melindungi AWS sumber daya Anda di bawah kunci kriptografi yang Anda hasilkan. Materi utama yang Anda impor dikaitkan dengan kunci KMS tertentu. Anda dapat mengimpor ulang bahan kunci yang sama ke kunci KMS yang sama, tetapi Anda tidak dapat mengimpor bahan kunci yang berbeda ke dalam kunci KMS dan Anda tidak dapat mengonversi kunci KMS yang dirancang untuk bahan kunci impor menjadi kunci KMS dengan bahan utama. AWS KMS

Pelajari lebih lanjut:

- the section called “Pilih spesifikasi kunci publik pembungkus”
- the section called “Pilih algoritme pembungkus”

Topik

Tentang material kunci yang diimpor

Sebelum Anda memutuskan untuk mengimpor bahan kunci ke dalam AWS KMS, Anda harus memahami karakteristik berikut dari bahan kunci impor.

Anda menghasilkan material kunci

Anda bertanggung jawab untuk menghasilkan materi utama menggunakan sumber keacakan yang memenuhi persyaratan keamanan Anda.

Anda dapat menghapus materi kunci

Anda dapat menghapus materi kunci yang diimpor dari kunci KMS, segera membuat kunci KMS tidak dapat digunakan. Selain itu, saat Anda mengimpor materi kunci ke kunci KMS, Anda dapat menentukan apakah kunci tersebut kedaluwarsa dan mengatur waktu kedaluwarsa. Ketika waktu kedaluwarsa tiba, AWS KMS hapus materi kunci. Tanpa materi kunci, kunci KMS tidak dapat digunakan dalam operasi kriptografi apa pun. Untuk mengembalikan kunci, Anda harus mengimpor ulang materi kunci yang sama ke dalam kunci.

Anda tidak dapat mengubah materi utama

Saat Anda mengimpor materi kunci ke kunci KMS, kunci KMS secara permanen dikaitkan dengan materi kunci tersebut. Anda dapat mengimpor ulang materi kunci yang sama, tetapi Anda tidak dapat mengimpor materi kunci yang berbeda ke dalam kunci KMS itu. Selain itu, Anda tidak dapat mengaktifkan rotasi tombol otomatis untuk kunci KMS dengan bahan kunci yang diimpor. Namun, Anda dapat memutar kunci KMS secara manual dengan bahan kunci yang diimpor.

Anda tidak dapat mengubah asal materi utama

Kunci KMS yang dirancang untuk bahan kunci impor memiliki nilai asal `EXTERNAL` yang tidak dapat diubah. Anda tidak dapat mengonversi kunci KMS untuk materi kunci yang diimpor untuk menggunakan materi kunci dari sumber lain, termasuk AWS KMS. Demikian pula, Anda tidak dapat mengonversi kunci KMS dengan bahan AWS KMS utama menjadi kunci yang dirancang untuk bahan kunci impor.

Anda tidak dapat mengekspor materi kunci

Anda tidak dapat mengekspor materi kunci apa pun yang Anda impor. AWS KMS tidak dapat mengembalikan materi kunci yang diimpor kepada Anda dalam bentuk apa pun. Anda harus menyimpan salinan materi kunci impor Anda di luar AWS, sebaiknya di pengelola kunci, seperti modul keamanan perangkat keras (HSM), sehingga Anda dapat mengimpor ulang materi kunci jika Anda menghapusnya atau kedaluwarsa.

Anda dapat membuat kunci Multi-wilayah dengan bahan kunci yang diimpor

Multi-Region dengan bahan kunci impor memiliki fitur kunci KMS dengan bahan kunci impor, dan dapat saling beroperasi di antaranya. Wilayah AWS Untuk membuat kunci Multi-region dengan materi kunci impor, Anda harus mengimpor bahan kunci yang sama ke kunci KMS primer dan ke setiap kunci replika. Lihat perinciannya di Mengimpor materi kunci ke kunci multi-Wilayah.

Tombol asimetris dan kunci HMAC portabel dan dapat dioperasikan

Anda dapat menggunakan bahan kunci asimetris dan bahan kunci HMAC di luar AWS untuk beroperasi dengan AWS KMS kunci dengan bahan kunci impor yang sama.

Berbeda dengan ciphertext AWS KMS simetris, yang terikat erat dengan kunci KMS yang digunakan dalam algoritma, AWS KMS menggunakan HMAC standar dan format asimetris untuk enkripsi, penandatanganan, dan pembuatan MAC. Akibatnya, kuncinya portabel dan mendukung skenario kunci escrow tradisional.

Ketika kunci KMS Anda telah mengimpor bahan kunci, Anda dapat menggunakan bahan kunci impor di luar AWS untuk melakukan operasi berikut.

- Kunci HMAC - Anda dapat memverifikasi tag HMAC yang dihasilkan oleh kunci HMAC KMS dengan bahan kunci yang diimpor. Anda juga dapat menggunakan kunci HMAC KMS dengan bahan kunci yang diimpor untuk memverifikasi tag HMAC yang dihasilkan oleh materi kunci di luar. AWS
- Kunci enkripsi asimetris — Anda dapat menggunakan kunci enkripsi asimetris pribadi Anda di luar AWS untuk mendekripsi ciphertext yang dienkripsi oleh kunci KMS dengan kunci publik yang sesuai. Anda juga dapat menggunakan kunci KMS asimetris Anda untuk mendekripsi ciphertext asimetris yang dihasilkan di luar. AWS
- Kunci penandatanganan asimetris — Anda dapat menggunakan kunci KMS penandatanganan asimetris dengan materi kunci yang diimpor untuk memverifikasi tanda tangan digital yang dihasilkan oleh kunci penandatanganan pribadi Anda di luar. AWS Anda juga dapat menggunakan kunci penandatanganan publik asimetris di luar AWS untuk memverifikasi tanda tangan yang dihasilkan oleh kunci KMS asimetris Anda.

Jika Anda mengimpor materi kunci yang sama ke kunci KMS yang berbeda dalam hal yang sama Wilayah AWS, kunci tersebut juga dapat dioperasikan. Untuk membuat kunci KMS yang dapat dioperasikan secara berbeda Wilayah AWS, buat kunci Multi-wilayah dengan bahan kunci yang diimpor.

Kunci enkripsi simetris tidak portabel atau interoperable

Ciphertext simetris yang AWS KMS menghasilkan tidak portabel atau interoperable. AWS KMS tidak mempublikasikan format ciphertext simetris yang dibutuhkan portabilitas, dan formatnya mungkin berubah tanpa pemberitahuan.

- AWS KMS tidak dapat mendekripsi ciphertext simetris yang Anda enkripsi di luar AWS, bahkan jika Anda menggunakan materi kunci yang telah Anda impor.
- AWS KMS tidak mendukung dekripsi ciphertext AWS KMS simetris apa pun di luar AWS KMS, bahkan jika ciphertext dienkripsi di bawah kunci KMS dengan materi kunci yang diimpor.
- Kunci KMS dengan bahan kunci impor yang sama tidak dapat dioperasikan. Ciphertext simetris yang AWS KMS menghasilkan ciphertext yang spesifik untuk setiap kunci KMS. Format ciphertext ini menjamin bahwa hanya kunci KMS yang data terenkripsi dapat mendekripsi itu.

Selain itu, Anda tidak dapat menggunakan AWS alat apa pun, seperti enkripsi sisi klien Amazon S3 AWS Encryption SDKatau Amazon S3, untuk mendekripsi ciphertext simetris. AWS KMS

Akibatnya, Anda tidak dapat menggunakan kunci dengan materi kunci yang diimpor untuk mendukung pengaturan escrow kunci di mana pihak ketiga yang berwenang dengan akses bersyarat ke materi kunci dapat mendekripsi ciphertext tertentu di luar. AWS KMS Untuk mendukung escrow kunci, gunakan AWS Encryption SDK untuk mengenkripsi pesan Anda di bawah kunci yang independen dari AWS KMS.

Anda bertanggung jawab atas ketersediaan dan daya tahan

AWS KMS dirancang untuk menjaga agar bahan kunci impor tetap tersedia. Tetapi AWS KMS tidak mempertahankan daya tahan bahan kunci impor pada tingkat yang sama dengan bahan utama yang AWS KMS menghasilkan. Lihat perinciannya di Melindungi material kunci yang diimpor.

Melindungi material kunci yang diimpor

Materi utama yang Anda impor dilindungi saat transit dan saat istirahat. Sebelum mengimpor materi kunci, Anda mengenkripsi (atau “membungkus”) materi kunci dengan kunci publik dari key pair RSA yang dihasilkan dalam modul keamanan AWS KMS perangkat keras (HSM) yang divalidasi di

bawah Program Validasi Modul Kriptografi FIPS 140-2. Anda dapat mengenkripsi materi kunci secara langsung dengan kunci publik pembungkus, atau mengenkripsi materi kunci dengan kunci simetris AES, dan kemudian mengenkripsi kunci simetris AES dengan kunci publik RSA.

Setelah diterima, AWS KMS dekripsi materi kunci dengan kunci pribadi yang sesuai di AWS KMS HSM dan mengenkripsi ulang di bawah kunci simetris AES yang hanya ada di memori volatile HSM. Materi kunci Anda tidak pernah meninggalkan HSM dalam teks biasa. Ini didekripsi hanya saat sedang digunakan dan hanya dalam AWS KMS HSM.

Penggunaan kunci KMS Anda dengan materi kunci impor ditentukan semata-mata oleh kebijakan kontrol akses yang Anda tetapkan pada kunci KMS. Selain itu, Anda dapat menggunakan alias dan tag untuk mengidentifikasi dan mengontrol akses ke kunci KMS. Anda dapat mengaktifkan dan menonaktifkan kunci, melihat dan mengedit propertinya, dan memantaunya menggunakan layanan seperti AWS CloudTrail.

Namun, Anda mempertahankan satu-satunya salinan failsafe dari materi kunci Anda. Sebagai imbalan atas ukuran kontrol ekstra ini, Anda bertanggung jawab atas daya tahan dan ketersediaan keseluruhan bahan kunci impor. AWS KMS dirancang untuk menjaga agar bahan kunci impor tetap tersedia. Tetapi AWS KMS tidak mempertahankan daya tahan bahan kunci impor pada tingkat yang sama dengan bahan utama yang AWS KMS menghasilkan.

Perbedaan daya tahan ini bermakna dalam kasus-kasus berikut:

- Saat Anda menetapkan waktu kedaluwarsa untuk materi kunci impor Anda, AWS KMS hapus materi kunci setelah kedaluwarsa. AWS KMS tidak menghapus kunci KMS atau metadata-nya. Anda dapat membuat CloudWatch alarm Amazon yang memberi tahu Anda saat materi kunci yang diimpor mendekati tanggal kedaluwarsanya.

  Anda tidak dapat menghapus materi kunci yang AWS KMS menghasilkan kunci KMS dan Anda tidak dapat mengatur materi AWS KMS kunci untuk kedaluwarsa, meskipun Anda dapat memutarnya.
- Saat Anda menghapus materi kunci yang diimpor secara manual, AWS KMS menghapus materi kunci tetapi tidak menghapus kunci KMS atau metadatanya. Sebaliknya, penjadwalan penghapusan kunci memerlukan masa tunggu 7 hingga 30 hari, setelah itu AWS KMS secara permanen menghapus kunci KMS, metadatanya, dan materi utamanya.
- Jika terjadi kegagalan di seluruh wilayah tertentu yang memengaruhi AWS KMS (seperti kehilangan daya total), AWS KMS tidak dapat secara otomatis mengembalikan materi kunci impor Anda. Namun, AWS KMS dapat mengembalikan kunci KMS dan metadata-nya.

Anda harus menyimpan salinan materi kunci yang diimpor di luar AWS dalam sistem yang Anda kontrol. Kami menyarankan Anda menyimpan salinan yang dapat diekspor dari bahan kunci yang diimpor dalam sistem manajemen kunci, seperti HSM. Jika materi kunci impor Anda dihapus atau kedaluwarsa, kunci KMS yang terkait menjadi tidak dapat digunakan sampai Anda mengimpor ulang materi kunci yang sama. Jika materi kunci impor Anda hilang secara permanen, ciphertext apa pun yang dienkripsi di bawah kunci KMS tidak dapat dipulihkan.

Izin untuk mengimpor material kunci

Untuk membuat dan mengelola kunci KMS dengan materi kunci yang diimpor, pengguna memerlukan izin untuk operasi dalam proses ini. Anda dapat memberikan`kms:GetParametersForImport`,`kms:ImportKeyMaterial`, dan `kms:DeleteImportedKeyMaterial` izin dalam kebijakan kunci saat Anda membuat kunci KMS. Di AWS KMS konsol, izin ini ditambahkan secara otomatis untuk administrator kunci saat Anda membuat kunci dengan asal materi kunci eksternal.

Untuk membuat kunci KMS dengan bahan kunci yang diimpor, prinsipal memerlukan izin berikut.

- kms: CreateKey (kebijakan IAM)
  - Untuk membatasi izin ini ke kunci KMS dengan materi kunci yang diimpor, gunakan kondisi KeyOrigin kebijakan kms: dengan nilai. `EXTERNAL`

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- kms: GetParametersForImport (Kebijakan utama atau kebijakan IAM)
  - Untuk membatasi izin ini pada permintaan yang menggunakan algoritme pembungkus tertentu dan spesifikasi kunci pembungkus, gunakan kondisi kebijakan kms: WrappingAlgorithm dan kms:. WrappingKeySpec
- kms: ImportKeyMaterial (Kebijakan utama atau kebijakan IAM)

- Untuk mengizinkan atau melarang materi utama yang kedaluwarsa dan mengontrol tanggal kedaluwarsa, gunakan ketentuan kebijakan kms: ExpirationModel dan kms:. ValidTo

Untuk mengimpor kembali materi kunci yang diimpor, prinsipal membutuhkan izin kms: GetParametersForImport dan kms:. ImportKeyMaterial

Untuk menghapus materi kunci yang diimpor, prinsipal membutuhkan DeleteImportedKeyMaterial izin kms:.

Misalnya, untuk memberikan `KMSAdminRole` izin contoh untuk mengelola semua aspek kunci KMS dengan materi kunci impor, sertakan pernyataan kebijakan kunci seperti berikut dalam kebijakan kunci KMS.

```
{
  "Sid": "Manage KMS keys with imported key material",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KMSAdminRole"
  },
  "Action": [
    "kms:GetParametersForImport",
    "kms:ImportKeyMaterial",
    "kms:DeleteImportedKeyMaterial"
  ]
}
```

Persyaratan untuk bahan kunci impor

Materi kunci yang Anda impor harus kompatibel dengan spesifikasi kunci kunci KMS terkait. Untuk pasangan kunci asimetris, impor hanya kunci pribadi pasangan. AWS KMS kunci publik berasal dari kunci privat.

AWS KMS mendukung spesifikasi kunci berikut untuk kunci KMS dengan bahan kunci yang diimpor.

Spesifikasi kunci kunci KMS	Persyaratan material utama
Kunci enkripsi simetris SYMMETRIC_DEFAULT	256-bit (32 byte) data biner

Spesifikasi kunci kunci KMS	Persyaratan material utama
	Di Wilayah China, itu harus berupa 128-bit (16 byte) data biner.
Kunci HMAC HMAC_224 HMAC_256 HMAC_384 HMAC_512	Materi kunci HMAC harus sesuai dengan RFC 2104. Panjang kunci harus sesuai dengan panjang yang ditentukan oleh spesifikasi kunci.
Kunci pribadi asimetris RSA RSA_2048 RSA_3072 RSA_4096	Kunci pribadi asimetris RSA yang Anda impor harus menjadi bagian dari key pair yang sesuai dengan RFC 3447. Modulus: 2048 bit, 3072 bit atau 4096 bit Jumlah bilangan prima: 2 (kunci RSA multi-prime tidak didukung) Materi kunci asimetris harus dikodekan BER atau dikodekan DER dalam format Public-Key Cryptography Standards (PKCS) #8 yang sesuai dengan RFC 5208.

Spesifikasi kunci kunci KMS	Persyaratan material utama
Kunci pribadi asimetris kurva elips ECC_NIST_P256 (secp256r1) ECC_NIST_P384 (secp384r1) ECC_NIST_P521 (secp521r1) ECC_SECG_P256K1 (secp256k1)	Kunci pribadi asimetris ECC yang Anda impor harus menjadi bagian dari key pair yang sesuai dengan RFC 5915. Kurva: NIST P-256, NIST P-384, NIST P-521, atau Secp256K1 Parameter: Kurva bernama saja (kunci ECC dengan parameter eksplisit ditolak) Koordinat titik publik: Dapat dikompresi, tidak dikompresi, atau proyektif Materi kunci asimetris harus dikodekan BER atau dikodekan DER dalam format Public-Key Cryptography Standards (PKCS) #8 yang sesuai dengan RFC 5208.
Kunci pribadi asimetris SM2 (hanya Wilayah China)	Kunci privat asimetris SM2 yang Anda impor harus menjadi bagian dari key pair yang sesuai dengan GM/T 0003. Kurva: SM2 Parameter: Kurva bernama saja (tombol SM2 dengan parameter eksplisit ditolak) Koordinat titik publik: Dapat dikompresi, tidak dikompresi, atau proyektif Materi kunci asimetris harus dikodekan BER atau dikodekan DER dalam format Public-Key Cryptography Standards (PKCS) #8 yang sesuai dengan RFC 5208.

Mengelola bahan kunci yang diimpor

Topik ini menjelaskan cara mengimpor dan mengimpor kembali materi kunci ke dalam kunci KMS dan cara membuat materi kunci impor yang kedaluwarsa secara otomatis.

Topik

Ikhtisar mengimpor bahan utama

Gambaran umum berikut menjelaskan cara mengimpor material kunci Anda ke dalam AWS KMS. Untuk detail selengkapnya tentang setiap langkah dalam proses, lihat topik yang sesuai.

1. Buat kunci KMS tanpa bahan kunci - Asal harus`EXTERNAL`. Asal kunci `EXTERNAL` menunjukkan bahwa kunci dirancang untuk bahan kunci yang diimpor dan AWS KMS mencegah menghasilkan bahan kunci untuk kunci KMS. Pada langkah selanjutnya Anda akan mengimpor materi kunci Anda sendiri ke kunci KMS ini.

   Materi kunci yang Anda impor harus kompatibel dengan spesifikasi kunci dari kunci terkait. AWS KMS Untuk informasi lebih lanjut tentang kompatibilitas, lihat. the section called “Persyaratan untuk bahan kunci impor”
2. Unduh kunci publik pembungkus dan token impor — Setelah menyelesaikan langkah 1, unduh kunci publik pembungkus dan token impor. Barang-barang ini melindungi materi utama Anda saat diimpor AWS KMS.

   Pada langkah ini, Anda memilih jenis (“spesifikasi kunci”) dari kunci pembungkus RSA dan algoritma pembungkus yang akan Anda gunakan untuk mengenkripsi data Anda dalam perjalanan. AWS KMS Anda dapat memilih spesifikasi kunci pembungkus dan algoritma kunci pembungkus yang berbeda setiap kali Anda mengimpor atau mengimpor ulang materi kunci yang sama.
3. Enkripsi materi kunci — Gunakan kunci publik pembungkus yang Anda unduh di langkah 2 untuk mengenkripsi materi kunci yang Anda buat di sistem Anda sendiri.

4. Mengimpor materi kunci — Unggah material kunci terenkripsi yang Anda buat pada langkah 3 dan token impor yang Anda unduh di langkah 2.

   Pada tahap ini, Anda dapat mengatur waktu kedaluwarsa opsional. Ketika materi kunci yang diimpor kedaluwarsa, AWS KMS menghapusnya, dan kunci KMS menjadi tidak dapat digunakan. Untuk terus menggunakan kunci KMS, Anda harus mengimpor ulang materi kunci yang sama.

   Ketika operasi impor selesai dengan sukses, status kunci kunci KMS berubah dari `PendingImport` ke. `Enabled` Anda sekarang dapat menggunakan kunci KMS dalam operasi kriptografi.

AWS KMS merekam entri di AWS CloudTrail log Anda saat Anda membuat kunci KMS, mengunduh kunci publik pembungkus dan token impor, dan mengimpor materi kunci. AWS KMS juga mencatat entri saat Anda menghapus materi kunci yang diimpor atau saat AWS KMS menghapus materi kunci yang kedaluwarsa.

Mengimpor ulang bahan utama

Jika Anda mengelola kunci KMS dengan materi kunci yang diimpor, Anda mungkin perlu mengimpor ulang materi kunci. Anda dapat mengimpor ulang materi kunci untuk menggantikan materi kunci yang kedaluwarsa atau dihapus, atau untuk mengubah model kedaluwarsa atau tanggal kedaluwarsa materi kunci.

Saat Anda mengimpor materi kunci ke kunci KMS, kunci KMS secara permanen dikaitkan dengan materi kunci tersebut. Anda dapat mengimpor ulang materi kunci yang sama, tetapi Anda tidak dapat mengimpor materi kunci yang berbeda ke dalam kunci KMS itu. Anda tidak dapat memutar materi kunci dan AWS KMS tidak dapat membuat materi kunci untuk kunci KMS dengan bahan kunci impor.

Anda dapat mengimpor ulang materi kunci kapan saja, pada jadwal apa pun yang memenuhi persyaratan keamanan Anda. Anda tidak perlu menunggu sampai materi kunci berada pada atau mendekati waktu kedaluwarsa.

Untuk mengimpor ulang material kunci, gunakan prosedur yang sama yang digunakan untuk mengimpor material kunci pertama kalinya, dengan pengecualian berikut.

- Gunakan kunci KMS yang ada, alih-alih membuat kunci KMS baru. Anda dapat melewatkan Langkah 1 dari prosedur impor.
- Saat Anda mengimpor ulang materi kunci, Anda dapat mengubah model kedaluwarsa dan tanggal kedaluwarsa.

Setiap kali Anda mengimpor materi kunci ke kunci KMS, Anda perlu mengunduh dan menggunakan kunci pembungkus baru dan token impor untuk kunci KMS. Prosedur pembungkus tidak memengaruhi konten materi utama, sehingga Anda dapat menggunakan kunci publik pembungkus yang berbeda dan algoritme pembungkus yang berbeda untuk mengimpor materi kunci yang sama.

Mengidentifikasi kunci KMS dengan bahan kunci impor

Ketika Anda membuat kunci KMS tanpa bahan kunci, nilai `Origin`properti kunci KMS adalah`EXTERNAL`, dan itu tidak dapat diubah. Berbeda dengan status kunci, `Origin` nilainya tidak bergantung pada ada atau tidaknya materi kunci.

Anda dapat menggunakan nilai `EXTERNAL` asal untuk mengidentifikasi kunci KMS yang dirancang untuk bahan kunci yang diimpor. Anda dapat menemukan asal kunci di AWS KMS konsol atau dengan menggunakan DescribeKeyoperasi. Anda juga dapat melihat properti dari material kunci, seperti apakah dan kapankah ini kedaluwarsa dengan menggunakan konsol atau API.

Untuk mengidentifikasi kunci KMS dengan bahan kunci impor (konsol)

1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Gunakan salah satu teknik berikut untuk melihat `Origin` properti kunci KMS Anda.
   - Untuk menambahkan kolom Origin ke tabel kunci KMS Anda, di sudut kanan atas, pilih ikon Pengaturan. Pilih Asal dan pilih Konfirmasi. Kolom Origin memudahkan untuk mengidentifikasi kunci KMS dengan nilai properti asal Eksternal (Import Key material).
   - Untuk menemukan nilai `Origin` properti kunci KMS tertentu, pilih ID kunci atau alias kunci KMS. Kemudian pilih tab Konfigurasi kriptografi. Tab ada di bawah bagian Konfigurasi umum.
4. Untuk melihat informasi mendetail tentang material kunci, pilih tab Material kunci. Tab ini muncul di halaman detail hanya untuk kunci KMS dengan materi kunci yang diimpor.

Untuk mengidentifikasi kunci KMS dengan material kunci impor (AWS KMS API)

Gunakan DescribeKeyoperasi. Respons mencakup `Origin` properti kunci KMS, model kedaluwarsa, dan tanggal kedaluwarsa, seperti yang ditunjukkan pada contoh berikut.

```
$  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"
        ]
    }
}
```

Membuat CloudWatch alarm untuk kedaluwarsa materi kunci yang diimpor

Anda dapat membuat CloudWatch alarm yang memberi tahu Anda ketika materi kunci yang diimpor dalam kunci KMS mendekati waktu kedaluwarsa. Misalnya, alarm dapat memberi tahu Anda ketika waktu kedaluwarsa kurang dari 30 hari lagi.

Saat Anda mengimpor materi kunci ke kunci KMS, Anda dapat secara opsional menentukan tanggal dan waktu saat materi kunci kedaluwarsa. Ketika materi kunci kedaluwarsa, AWS KMS menghapus materi kunci dan kunci KMS menjadi tidak dapat digunakan. Untuk menggunakan kunci KMS lagi, Anda harus mengimpor ulang materi kunci. Namun, jika Anda mengimpor ulang materi kunci sebelum kedaluwarsa, Anda dapat menghindari gangguan proses yang menggunakan kunci KMS tersebut.

Alarm ini menggunakan `SecondsUntilKeyMaterialExpires`metrik yang AWS KMS diterbitkan CloudWatch untuk kunci KMS dengan materi kunci impor yang kedaluwarsa. Setiap alarm menggunakan metrik ini untuk memantau bahan kunci yang diimpor untuk kunci KMS tertentu. Anda tidak dapat membuat alarm tunggal untuk semua kunci KMS dengan materi kunci kedaluwarsa atau alarm untuk kunci KMS yang mungkin Anda buat di masa depan.

Persyaratan

Sumber daya berikut diperlukan untuk CloudWatch alarm yang memantau berakhirnya bahan kunci yang diimpor.

- Kunci KMS dengan bahan kunci impor yang kedaluwarsa. Untuk bantuan, lihat Mengidentifikasi kunci KMS dengan bahan kunci impor.
- Topik Amazon SNS. Untuk detailnya, lihat Membuat topik Amazon SNS di CloudWatch Panduan Pengguna Amazon.

Buat alarm

Ikuti petunjuk di Buat CloudWatch alarm berdasarkan ambang statis menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pilih metrik	Pilih KMS, lalu pilih Metrik Per-Key. Pilih baris dengan tombol KMS dan `SecondsUntilKeyMaterialExpires` metrik. Kemudian pilih Pilih metrik. Daftar Metrik menampilkan `SecondsUntilKeyMaterialExpires` metrik hanya untuk kunci KMS dengan materi kunci impor yang kedaluwarsa. Jika Anda tidak memiliki kunci KMS dengan properti ini di akun dan Wilayah, daftar ini kosong.
Statistik	Minimum
Periode	1 menit
Jenis ambang	Statis
Kapanpun...	Setiap kali ***nama metrik lebih*** besar dari 1

Menghapus material kunci yang diimpor

Anda dapat menghapus materi kunci yang diimpor dari kunci KMS kapan saja. Juga, ketika materi kunci yang diimpor dengan tanggal kedaluwarsa kedaluwarsa, AWS KMS menghapus materi kunci. Dalam kedua kasus, ketika materi kunci dihapus, status kunci kunci KMS berubah menjadi impor

tertunda, dan kunci KMS tidak dapat digunakan dalam operasi kriptografi apa pun sampai Anda mengimpor ulang materi kunci yang sama. (Anda tidak dapat mengimpor materi kunci lainnya ke dalam kunci KMS.)

Seiring dengan menonaktifkan kunci KMS dan menarik izin, menghapus materi kunci dapat digunakan sebagai strategi untuk dengan cepat, tetapi untuk sementara, menghentikan penggunaan kunci KMS. Sebaliknya, penjadwalan penghapusan kunci KMS dengan bahan kunci yang diimpor juga dengan cepat menghentikan penggunaan kunci KMS. Namun, jika penghapusan tidak dibatalkan selama masa tunggu, kunci KMS, materi kunci, dan semua metadata kunci akan dihapus secara permanen. Lihat perinciannya di the section called “Menghapus kunci KMS dengan materi kunci yang diimpor”.

Untuk menghapus materi kunci, Anda dapat menggunakan AWS KMS konsol atau operasi DeleteImportedKeyMaterialAPI. AWS KMS mencatat entri di AWS CloudTrail log Anda saat Anda menghapus materi kunci yang diimpor dan saat AWS KMS menghapus materi kunci yang kedaluwarsa.

Topik

- Bagaimana menghapus materi kunci memengaruhi layanan AWS
- Hapus material kunci (konsol)
- Hapus materi kunci (AWS KMS API)

Bagaimana menghapus materi kunci memengaruhi layanan AWS

Saat Anda menghapus materi kunci, kunci KMS tanpa materi kunci menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhi Layanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Lihat perinciannya di Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Hapus material kunci (konsol)

Anda dapat menggunakan AWS Management Console untuk menghapus materi kunci.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.

3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Lakukan salah satu hal berikut ini:
   - Pilih kotak centang untuk kunci KMS dengan bahan kunci impor. Pilih Tindakan kunci, Hapus material kunci.
   - Pilih alias atau ID kunci kunci KMS dengan materi kunci impor. Pilih tab Material kunci kemudian pilih Hapus material kunci.
5. Konfirmasi bahwa Anda ingin menghapus material kunci, lalu pilih Hapus material kunci. Status kunci KMS, yang sesuai dengan status kuncinya, berubah menjadi Impor tertunda.

Hapus materi kunci (AWS KMS API)

Untuk menggunakan AWS KMS API untuk menghapus materi kunci, kirim DeleteImportedKeyMaterialpermintaan. Contoh berikut ini menunjukkan cara melakukan ini dengan AWS CLI.

Ganti *1234abcd-12ab-34cd-56ef-1234567890ab* dengan ID kunci kunci KMS yang materi utamanya ingin Anda hapus. Anda dapat menggunakan ID kunci KMS atau ARN tetapi Anda tidak dapat menggunakan alias untuk operasi ini.

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

Menghapus kunci KMS dengan materi kunci yang diimpor

Menghapus bahan kunci kunci KMS dengan bahan kunci impor bersifat sementara dan dapat dibalik. Untuk mengembalikan kunci, impor kembali materi utamanya.

Sebaliknya, menghapus kunci KMS tidak dapat diubah. Jika Anda menjadwalkan penghapusan kunci dan masa tunggu yang diperlukan berakhir, AWS KMS secara permanen dan permanen menghapus kunci KMS, materi utamanya, dan semua metadata yang terkait dengan kunci KMS.

Namun, risiko dan konsekuensi dari menghapus kunci KMS dengan bahan kunci yang diimpor tergantung pada jenis (“spesifikasi kunci”) dari kunci KMS.

- Kunci enkripsi simetris — Jika Anda menghapus kunci KMS enkripsi simetris, semua ciphertext yang tersisa yang dienkripsi oleh kunci tersebut tidak dapat dipulihkan. Anda tidak dapat membuat kunci KMS enkripsi simetris baru yang dapat mendekripsi ciphertext dari kunci KMS enkripsi simetris yang dihapus, bahkan jika Anda memiliki materi kunci yang sama. Metadata unik untuk

setiap kunci KMS terikat secara kriptografis ke setiap ciphertext simetris. Fitur keamanan ini menjamin bahwa hanya kunci KMS yang mengenkripsi ciphertext simetris yang dapat mendekripsi, tetapi mencegah Anda membuat ulang kunci KMS yang setara.

- Kunci asimetris dan HMAC — Jika Anda memiliki materi kunci asli, Anda dapat membuat kunci KMS baru dengan properti kriptografi yang sama dengan kunci KMS asimetris atau HMAC yang telah dihapus. AWS KMS menghasilkan ciphertext dan tanda tangan RSA standar, tanda tangan ECC, dan tag HMAC, yang tidak menyertakan fitur keamanan unik apa pun. Selain itu, Anda dapat menggunakan kunci HMAC atau kunci pribadi dari asymmetric key pair di luar. AWS

  Kunci KMS baru yang Anda buat dengan bahan kunci asimetris atau HMAC yang sama akan memiliki pengenal kunci yang berbeda. Anda harus membuat kebijakan kunci baru, membuat ulang alias apa pun, dan memperbarui kebijakan dan hibah IAM yang ada untuk merujuk ke kunci baru.

Mengimpor bahan kunci langkah 1: Buat materi AWS KMS key tanpa kunci

Secara default, AWS KMS buat materi kunci untuk Anda saat Anda membuat kunci KMS. Untuk mengimpor materi kunci Anda sendiri, mulailah dengan membuat kunci KMS tanpa materi kunci. Kemudian impor bahan kunci. Untuk membuat kunci KMS tanpa bahan kunci, gunakan AWS KMS konsol atau CreateKeyoperasi.

Untuk membuat kunci tanpa bahan kunci, tentukan asalEXTERNAL. Properti asal kunci KMS tidak dapat diubah. Setelah Anda membuatnya, Anda tidak dapat mengonversi kunci KMS yang dirancang untuk materi kunci yang diimpor menjadi kunci KMS dengan bahan kunci dari AWS KMS atau sumber lainnya.

Status kunci dari kunci KMS dengan `EXTERNAL` asal dan tidak ada `PendingImport` bahan kunci. Kunci KMS dapat tetap dalam `PendingImport` keadaan tanpa batas waktu. Namun, Anda tidak dapat menggunakan kunci KMS dalam `PendingImport` keadaan dalam operasi kriptografi. Saat Anda mengimpor materi kunci, status kunci kunci KMS berubah menjadi`Enabled`, dan Anda dapat menggunakannya dalam operasi kriptografi.

AWS KMSmerekam peristiwa di AWS CloudTrail log Anda saat Anda membuat kunci KMS, mengunduh kunci publik dan token impor, dan mengimpor materi kunci. AWS KMSjuga mencatat CloudTrail peristiwa saat Anda menghapus materi kunci yang diimpor atau saat AWS KMS menghapus materi kunci yang kedaluwarsa.

Untuk informasi tentang cara membuat kunci multi-wilayah dengan material kunci yang diimpor, lihat Mengimpor materi kunci ke kunci multi-Wilayah.

Topik

- Membuat kunci KMS tanpa bahan kunci (konsol)
- Membuat kunci KMS tanpa materi kunci (AWS KMSAPI)

Membuat kunci KMS tanpa bahan kunci (konsol)

Anda hanya perlu membuat kunci KMS untuk materi kunci yang diimpor sekali. Anda dapat mengimpor dan mengimpor kembali materi kunci yang sama ke kunci KMS yang ada sesering yang Anda butuhkan, tetapi Anda tidak dapat mengimpor materi kunci yang berbeda ke dalam kunci KMS. Untuk detailnya, lihat Langkah 2: Unduh kunci publik pembungkus dan token impor.

Untuk menemukan kunci KMS yang ada dengan materi kunci yang diimpor di tabel kunci yang dikelola Pelanggan Anda, gunakan ikon roda gigi di sudut kanan atas untuk menampilkan kolom Asal dalam daftar kunci KMS. Kunci yang diimpor memiliki nilai Asal Eksternal (Bahan Kunci Impor).

Untuk membuat kunci KMS dengan bahan kunci impor, mulailah dengan mengikuti petunjuk dasar untuk membuat kunci KMS dari jenis kunci pilihan Anda, dengan pengecualian berikut.

Setelah memilih penggunaan kunci, lakukan hal berikut:

1. Perluas Opsi lanjutan.
2. Untuk asal bahan utama, pilih Eksternal (Impor bahan kunci).
3. Pilih kotak centang di sebelah Saya memahami implikasi keamanan dan daya tahan menggunakan kunci impor untuk menunjukkan bahwa Anda memahami implikasi penggunaan bahan kunci impor. Untuk membaca tentang implikasi ini, lihat Melindungi material kunci yang diimpor.
4. Kembali ke instruksi dasar. Langkah-langkah yang tersisa dari prosedur dasar adalah sama untuk semua kunci KMS dari jenis itu.

Ketika Anda memilih Selesai, Anda telah membuat kunci KMS tanpa materi kunci dan status (status kunci) dari impor Tertunda.

Namun, alih-alih kembali ke tabel kunci terkelola Pelanggan, konsol menampilkan halaman tempat Anda dapat mengunduh kunci publik dan mengimpor token yang Anda perlukan untuk mengimpor materi kunci Anda. Anda dapat melanjutkan dengan langkah download sekarang, atau pilih Cancel untuk berhenti pada titik ini. Anda dapat kembali ke langkah pengunduhan ini kapan saja.

Selanjutnya: Langkah 2: Unduh kunci publik pembungkus dan token impor.

Membuat kunci KMS tanpa materi kunci (AWS KMSAPI)

Untuk menggunakan AWS KMSAPI untuk membuat kunci KMS enkripsi simetris tanpa bahan kunci, kirim CreateKeypermintaan dengan `Origin` parameter yang disetel ke. `EXTERNAL` Contoh berikut ini menunjukkan cara melakukan dengan AWS Command Line Interface (AWS CLI).

```
$ aws kms create-key --origin EXTERNAL
```

Ketika perintah berhasil, Anda melihat output yang serupa dengan berikut ini. AWS KMSKuncinya `Origin` adalah `EXTERNAL` dan `KeyState` itu adalah`PendingImport`.

> **Tip**
>
> Jika perintah tidak berhasil, Anda mungkin melihat a `KMSInvalidStateException` atau a`NotFoundException`. Anda dapat mencoba kembali permintaan tersebut.

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

Salin `KeyId` nilai dari output perintah Anda untuk digunakan di langkah selanjutnya, dan kemudian lanjutkan keLangkah 2: Unduh kunci publik pembungkus dan token impor.

> **Note**
>
> Perintah ini menciptakan kunci KMS enkripsi simetris dengan `KeySpec` dari `SYMMETRIC_DEFAULT` dan `KeyUsage` dari. `ENCRYPT_DECRYPT` Anda dapat menggunakan parameter opsional `--key-spec` dan `--key-usage` untuk membuat kunci KMS asimetris atau HMAC. Untuk informasi lebih lanjut, lihat CreateKeyoperasi.

Mengimpor materi kunci langkah 2: Unduh kunci publik pembungkus dan token impor

Setelah Anda membuat materi AWS KMS key tanpa kunci, unduh kunci publik pembungkus dan token impor untuk kunci KMS tersebut dengan menggunakan AWS KMS konsol atau API. GetParametersForImport Kunci publik pembungkus dan token impor adalah set tak terpisahkan yang harus digunakan bersama.

Anda akan menggunakan kunci publik pembungkus untuk mengenkripsi materi utama Anda untuk transportasi. Sebelum mengunduh RSA wrapping key pair, Anda memilih length (key spec) dari RSA wrapping key pair dan algoritma wrapping yang akan Anda gunakan untuk mengenkripsi material kunci impor Anda untuk diangkut pada langkah 3. AWS KMS juga mendukung spesifikasi kunci pembungkus SM2 (hanya Wilayah China).

Setiap pembungkus kunci publik dan set token impor berlaku selama 24 jam. Jika Anda tidak menggunakannya untuk mengimpor materi kunci dalam waktu 24 jam setelah mengunduhnya, Anda harus mengunduh set baru. Anda dapat mengunduh kunci publik pembungkus baru dan mengimpor set token kapan saja. Ini memungkinkan Anda mengubah panjang kunci pembungkus RSA Anda (“spesifikasi kunci”) atau mengganti set yang hilang.

Anda juga dapat mengunduh kunci publik pembungkus dan set token impor untuk mengimpor kembali materi kunci yang sama ke kunci KMS. Anda dapat melakukan ini untuk mengatur atau mengubah waktu kedaluwarsa untuk materi kunci, atau untuk memulihkan materi kunci yang kedaluwarsa atau dihapus. Anda harus mengunduh dan mengenkripsi ulang materi kunci Anda setiap kali Anda mengimpornya. AWS KMS

Penggunaan kunci publik pembungkus

Unduhan menyertakan kunci publik yang unik untuk Anda Akun AWS, juga disebut kunci publik pembungkus.

Sebelum Anda mengimpor materi kunci, Anda mengenkripsi materi kunci dengan kunci pembungkus publik, dan kemudian mengunggah materi kunci terenkripsi ke. AWS KMS Saat AWS KMS menerima materi kunci terenkripsi Anda, ia mendekripsi materi kunci dengan kunci pribadi yang sesuai, kemudian mengenkripsi ulang materi kunci di bawah kunci simetris AES, semuanya dalam modul keamanan perangkat keras (HSM). AWS KMS

Penggunaan token impor

Unduhan menyertakan token impor dengan metadata yang memastikan bahwa materi kunci Anda diimpor dengan benar. Saat Anda mengunggah materi kunci terenkripsi AWS KMS, Anda harus mengunggah token impor yang sama dengan yang Anda unduh di langkah ini.

Pilih spesifikasi kunci publik pembungkus

Untuk melindungi materi kunci Anda selama impor, Anda mengenkripsinya menggunakan kunci publik pembungkus yang Anda unduh AWS KMS, dan algoritme pembungkus yang didukung. Anda memilih spesifikasi kunci sebelum mengunduh kunci publik pembungkus dan token impor. Semua pasangan kunci pembungkus dihasilkan dalam modul keamanan AWS KMS perangkat keras (HSM). Kunci pribadi tidak pernah meninggalkan HSM dalam teks biasa.

Spesifikasi kunci pembungkus RSA

Spesifikasi kunci dari kunci publik pembungkus menentukan panjang kunci dalam key pair RSA yang melindungi material kunci Anda selama pengangkutannya. AWS KMS Secara umum, kami sarankan menggunakan kunci publik pembungkus terpanjang yang praktis. Kami menawarkan beberapa spesifikasi kunci publik yang lengkap untuk mendukung berbagai HSM dan manajer kunci.

AWS KMS mendukung spesifikasi kunci berikut untuk kunci pembungkus RSA yang digunakan untuk mengimpor bahan kunci dari semua jenis, kecuali seperti yang disebutkan.

- RSA_4096 (lebih disukai)
- RSA_3072
- RSA_2048

> **Note**
>
> Kombinasi berikut TIDAK didukung: bahan kunci ECC_NIST_P521, spesifikasi kunci pembungkus publik RSA_2048, dan algoritma pembungkus RSAES_OAEP_SHA_*.

Anda tidak dapat langsung membungkus materi kunci ECC_NIST_P521 dengan kunci pembungkus publik RSA_2048. Gunakan kunci pembungkus yang lebih besar atau algoritma pembungkus RSA_AES_KEY_WRAP_SHA_*.

Spesifikasi kunci pembungkus SM2 (khusus Wilayah China)

AWS KMS mendukung spesifikasi kunci berikut untuk kunci pembungkus SM2 yang digunakan untuk mengimpor bahan kunci asimetris.

- SM2

Pilih algoritme pembungkus

Untuk melindungi materi kunci Anda selama impor, Anda mengenkripsinya menggunakan kunci publik pembungkus yang diunduh dan algoritme pembungkus yang didukung.

AWS KMS mendukung beberapa algoritma pembungkus RSA standar dan algoritma pembungkus hibrida dua langkah. Secara umum, sebaiknya gunakan algoritme pembungkus paling aman yang kompatibel dengan bahan kunci impor dan spesifikasi kunci pembungkus Anda. Biasanya, Anda memilih algoritme yang didukung oleh modul keamanan perangkat keras (HSM) atau sistem manajemen kunci yang melindungi material kunci Anda.

Tabel berikut menunjukkan algoritma pembungkus yang didukung untuk setiap jenis bahan kunci dan kunci KMS. Algoritma tercantum dalam urutan preferensi.

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
Kunci enkripsi simetris Kunci AES 256-bit Kunci SM4 128-bit (hanya Wilayah China)	Algoritma pembungkus: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Algoritma pembungkus yang tidak digunakan lagi: RSAES_PKCS1_V1

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
	Note Per 10 Oktober 2023, AWS KMS tidak mendukung algoritma pembungkus RSAES_PKCS1_V1_5. Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096
Kunci pribadi RSA asimetris	Algoritma pembungkus: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE (hanya Wilayah China) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah China)

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
Kunci pribadi kurva elips asimetris (ECC) Anda tidak dapat menggunakan algoritma pembungkus RSAES_OAEP_SHA_* dengan spesifikasi kunci pembungkus RSA_2048 untuk membungkus materi kunci ECC_NIST_P521.	Algoritma pembungkus: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (hanya Wilayah China) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah China)
Kunci pribadi SM2 asimetris (hanya Wilayah China)	Algoritma pembungkus: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (hanya Wilayah China) Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096 SM2 (Hanya Wilayah China)

Material kunci	Algoritma dan spesifikasi pembungkus yang didukung
Kunci HMAC	Algoritma pembungkus: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Spesifikasi kunci pembungkus: RSA_2048 RSA_3072 RSA_4096

> **Note**
>
> Algoritma `RSA_AES_KEY_WRAP_SHA_256` dan `RSA_AES_KEY_WRAP_SHA_1` pembungkus tidak didukung di Wilayah China.

- `RSA_AES_KEY_WRAP_SHA_256`— Algoritma pembungkus hibrida dua langkah yang menggabungkan enkripsi materi kunci Anda dengan kunci simetris AES yang Anda hasilkan, dan kemudian mengenkripsi kunci simetris AES dengan kunci pembungkus publik RSA yang diunduh dan algoritma pembungkus RSAES_OAEP_SHA_256.

  Algoritma `RSA_AES_KEY_WRAP_SHA_*` pembungkus diperlukan untuk membungkus materi kunci pribadi RSA, kecuali di Wilayah China, di mana Anda harus menggunakan algoritma pembungkus. `SM2PKE`
- `RSA_AES_KEY_WRAP_SHA_1`— Algoritma pembungkus hibrida dua langkah yang menggabungkan enkripsi materi kunci Anda dengan kunci simetris AES yang Anda hasilkan, dan kemudian mengenkripsi kunci simetris AES dengan kunci publik pembungkus RSA yang diunduh dan algoritma pembungkus RSAES_OAEP_SHA_1.

  Algoritma `RSA_AES_KEY_WRAP_SHA_*` pembungkus diperlukan untuk membungkus materi kunci pribadi RSA, kecuali di Wilayah China, di mana Anda harus menggunakan algoritma pembungkus. `SM2PKE`

- `RSAES_OAEP_SHA_256`— Algoritma enkripsi RSA dengan Optimal Asymmetric Encryption Padding (OAEP) dengan fungsi hash SHA-256.
- `RSAES_OAEP_SHA_1`— Algoritma enkripsi RSA dengan Optimal Asymmetric Encryption Padding (OAEP) dengan fungsi hash SHA-1.
- `RSAES_PKCS1_V1_5`(Usang; per 10 Oktober 2023, AWS KMS tidak mendukung algoritma pembungkus RSAES_PKCS1_V1_5) — Algoritma enkripsi RSA dengan format padding yang ditentukan dalam PKCS #1 Versi 1.5.
- `SM2PKE`(Hanya Wilayah China) — Algoritma enkripsi berbasis kurva elips yang ditentukan oleh OSCCA dalam GM/T 0003.4-2012.

Topik

Mengunduh kunci publik pembungkus dan token impor (konsol)

Anda dapat menggunakan AWS KMS konsol untuk mengunduh kunci publik pembungkus dan token impor.

1. Jika Anda baru saja menyelesaikan langkah-langkah untuk membuat kunci KMS tanpa materi kunci dan Anda berada di tombol pembungkus Unduh dan halaman token impor, lewati ke. Step 9
2. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
3. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
4. Di panel navigasi, pilih Kunci yang dikelola pelanggan.

> **Tip**
>
> Anda dapat mengimpor materi kunci hanya ke kunci KMS dengan Origin of External (Import key material). Ini menunjukkan bahwa kunci KMS dibuat tanpa bahan kunci. Untuk menambahkan kolom Asal ke tabel Anda, di sudut kanan atas halaman, pilih ikon pengaturan

().
Hidupkan Asal, lalu pilih Konfirmasi.

5. Pilih alias atau ID kunci kunci KMS yang menunggu impor.
6. Pilih tab Konfigurasi kriptografi dan lihat nilainya. Tab ada di bawah bagian Konfigurasi umum.

   Anda hanya dapat mengimpor materi kunci ke kunci KMS Asal Eksternal (bahan Kunci Impor). Untuk informasi tentang membuat kunci KMS dengan materi kunci impor, lihat,Mengimpor bahan kunci untuk AWS KMS kunci.
7. Pilih tab Materi kunci dan kemudian pilih Impor bahan kunci.

   Tab Materi kunci hanya muncul untuk kunci KMS yang memiliki nilai Asal Eksternal (bahan Import Key).
8. Untuk Pilih spesifikasi tombol pembungkus, pilih konfigurasi untuk kunci KMS Anda. Setelah Anda membuat kunci ini, Anda tidak dapat mengubah spesifikasi kunci.
9. Untuk Memilih algoritme pembungkus, pilih opsi yang akan Anda gunakan untuk mengenkripsi material kunci Anda. Untuk informasi selengkapnya tentang opsi, lihat Pilih Algoritme Pembungkus.
10. Pilih Download wrapping public key dan import token, lalu simpan file.

    Jika Anda memiliki opsi Selanjutnya, untuk melanjutkan proses sekarang, pilih Selanjutnya. Untuk melanjutkan nanti, pilih Batalkan.
11. Dekompresi file `.zip` yang Anda simpan pada langkah sebelumnya (`Import_Parameters_<key_id>_<timestamp>`).

    Folder berisi file-file berikut:

    - Kunci publik pembungkus dalam file bernama`WrappingPublicKey.bin`.
    - Token impor dalam file bernama`ImportToken.bin`.
    - Sebuah file teks bernama README.txt. File ini berisi informasi tentang kunci publik pembungkus, algoritma pembungkus yang digunakan untuk mengenkripsi materi kunci Anda, dan tanggal dan waktu ketika kunci publik pembungkus dan token impor kedaluwarsa.
12. Untuk melanjutkan proses, lihat mengenkripsi material kunci Anda.

Mengunduh kunci publik pembungkus dan token impor (AWS KMS API)

Untuk mengunduh kunci publik dan token impor, gunakan GetParametersForImportAPI. Tentukan kunci KMS yang akan dikaitkan dengan materi kunci yang diimpor. Kunci KMS ini harus memiliki nilai Origin. `EXTERNAL`

Contoh ini menentukan algoritma `RSA_AES_KEY_WRAP_SHA_256` pembungkus, spesifikasi kunci publik pembungkus RSA_3072, dan ID kunci contoh. Ganti nilai contoh ini dengan nilai yang valid untuk unduhan Anda. Untuk ID kunci, Anda dapat menggunakan ID kunci atau kunci ARN, tetapi Anda tidak dapat menggunakan nama alias atau alias ARN dalam operasi ini.

```
$ 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
```

Ketika perintah berhasil, Anda melihat output yang serupa dengan berikut ini:

```
{
    "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)"
}
```

Untuk menyiapkan data untuk langkah berikutnya, base64 memecahkan kode kunci publik dan token impor dan menyimpan nilai yang diterjemahkan dalam file.

Untuk base64 memecahkan kode kunci publik dan token impor:

1. Salin kunci publik yang dikodekan base64 (diwakili oleh *kunci publik (dikodekan base64)* dalam output contoh), tempelkan ke file baru, lalu simpan file tersebut. Berikan file nama deskriptif, seperti`PublicKey.b64`.
2. Gunakan OpenSSL ke base64 mendekodekan isi file dan menyimpan data yang didekodekan ke file baru. Contoh berikut mendekodekan data dalam file yang Anda simpan di langkah sebelumnya (`PublicKey.b64`) dan menyimpan output ke file baru bernama `WrappingPublicKey.bin`.

```
$ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
```

3. Salin token impor yang dikodekan base64 (diwakili oleh *token impor (dikodekan base64)* dalam contoh keluaran), tempel ke file baru, lalu simpan file tersebut. Berikan nama deskriptif pada file, misalnya `importtoken.b64`.
4. Gunakan OpenSSL ke base64 mendekodekan isi file dan menyimpan data yang didekodekan ke file baru. Contoh berikut mendekodekan data dalam file yang Anda simpan di langkah sebelumnya (`ImportToken.b64`) dan menyimpan output ke file baru bernama `ImportToken.bin`.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Lanjut ke Langkah 3: Enkripsi material kunci.

Mengimpor material kunci langkah 3: Enkripsi material kunci

Setelah Anda mengunduh kunci publik dan token impor, enkripsi materi kunci Anda menggunakan kunci publik yang Anda unduh dan algoritme pembungkus yang Anda tentukan. Jika Anda perlu mengganti kunci publik atau token impor, atau mengubah algoritma pembungkus, Anda harus mengunduh kunci publik baru dan token impor. Untuk informasi tentang kunci publik dan algoritma pembungkus yang AWS KMS mendukung, lihat Pilih spesifikasi kunci publik pembungkus dan. Pilih algoritme pembungkus

Material kunci harus dalam format biner. Untuk detail informasi, lihat Persyaratan untuk bahan kunci impor.

> **Note**
>
> Untuk pasangan kunci asimetris, enkripsi dan impor hanya kunci pribadi. AWS KMS kunci publik berasal dari kunci privat.
> Kombinasi berikut TIDAK didukung: bahan kunci ECC_NIST_P521, spesifikasi kunci pembungkus publik RSA_2048, dan algoritma pembungkus RSAES_OAEP_SHA_*.
> Anda tidak dapat langsung membungkus materi kunci ECC_NIST_P521 dengan kunci pembungkus publik RSA_2048. Gunakan kunci pembungkus yang lebih besar atau algoritma pembungkus RSA_AES_KEY_WRAP_SHA_*.

Algoritma pembungkus RSA_AES_KEY_WRAP_SHA_256 dan RSA_AES_KEY_WRAP_SHA_1 tidak didukung di Wilayah China.

Biasanya, Anda mengenkripsi material kunci Anda ketika Anda mengekspornya dari modul keamanan perangkat keras (HSM) atau sistem manajemen kunci. Untuk informasi tentang cara mengekspor material kunci dalam format biner, lihat dokumentasi untuk HSM atau sistem manajemen kunci Anda. Anda juga dapat merujuk ke bagian berikut yang memberikan bukti konsep demonstrasi menggunakan OpenSSL.

Saat Anda mengenkripsi materi kunci Anda, gunakan algoritme pembungkus yang sama dengan yang Anda tentukan saat mengunduh kunci publik dan token impor. Untuk menemukan algoritma pembungkus yang Anda tentukan, lihat peristiwa CloudTrail log untuk GetParametersForImportpermintaan terkait.

Hasilkan bahan utama untuk pengujian

Perintah OpenSSL berikut menghasilkan material kunci dari setiap jenis yang didukung untuk pengujian. Contoh-contoh ini disediakan hanya untuk pengujian dan proof-of-concept demonstrasi. Untuk sistem produksi, gunakan metode yang lebih aman untuk menghasilkan materi utama Anda, seperti modul keamanan perangkat keras atau sistem manajemen kunci.

Untuk mengonversi kunci pribadi pasangan kunci asimetris menjadi format yang dikodekan DER, salurkan perintah pembuatan material kunci ke perintah berikut. `openssl pkcs8 topk8`Parameter mengarahkan OpenSSL untuk mengambil kunci pribadi sebagai input dan mengembalikan kunci berformat PKCS #8. (Perilaku default adalah sebaliknya.)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

Perintah berikut menghasilkan bahan kunci uji untuk setiap jenis kunci yang didukung.

- Kunci enkripsi simetris (32 byte)

  Perintah ini menghasilkan kunci simetris 256-bit (string acak 32-byte) dan menyimpannya dalam file. `PlaintextKeyMaterial.bin` Anda tidak perlu menyandikan materi kunci ini.

  ```
  openssl rand -out PlaintextKeyMaterial.bin 32
  ```

  Hanya di Wilayah China, Anda harus menghasilkan kunci simetris 128-bit (string acak 16-byte).

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- Kunci HMAC

  Perintah ini menghasilkan string byte acak dari ukuran yang ditentukan. Anda tidak perlu menyandikan materi kunci ini.

  Panjang kunci HMAC Anda harus sesuai dengan panjang yang ditentukan oleh spesifikasi kunci kunci KMS. Misalnya, jika kunci KMS adalah HMAC_384, Anda harus mengimpor kunci 384-bit (48-byte).

```
openssl rand -out HMAC_224_PlaintextKey.bin 28

openssl rand -out HMAC_256_PlaintextKey.bin 32

openssl rand -out HMAC_384_PlaintextKey.bin 48

openssl rand -out HMAC_512_PlaintextKey.bin 64
```

- Kunci pribadi RSA

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_2048_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:3072 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_3072_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_4096_PrivateKey.der
```

- Kunci pribadi ECC

```
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
```

- Kunci pribadi SM2 (hanya Wilayah China)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -
outform der -nocrypt > SM2_PrivateKey.der
```

Contoh mengenkripsi materi kunci dengan OpenSSL

Contoh berikut menunjukkan cara menggunakan OpenSSL untuk mengenkripsi materi kunci Anda dengan kunci publik yang Anda unduh. Untuk mengenkripsi materi kunci Anda menggunakan kunci publik SM2 (khusus Wilayah China), gunakan kelas. `SM2OfflineOperationHelper`

> **Important**
>
> Contoh-contoh ini adalah bukti demonstrasi konsep saja. Untuk sistem produksi, gunakan metode yang lebih aman (seperti HSM komersial atau sistem manajemen kunci) untuk menghasilkan dan menyimpan material kunci Anda.
> Kombinasi berikut TIDAK didukung: bahan kunci ECC_NIST_P521, spesifikasi kunci pembungkus publik RSA_2048, dan algoritma pembungkus RSAES_OAEP_SHA_*.
> Anda tidak dapat langsung membungkus materi kunci ECC_NIST_P521 dengan kunci pembungkus publik RSA_2048. Gunakan kunci pembungkus yang lebih besar atau algoritma pembungkus RSA_AES_KEY_WRAP_SHA_*.

RSAES_OAEP_SHA_1

AWS KMS mendukung RSAES_OAEP_SHA_1 untuk kunci enkripsi simetris (SYMMETRIC_DEFAULT), kunci pribadi kurva elips (ECC), kunci pribadi SM2, dan kunci HMAC.

RSAES_OAEP_SHA_1 tidak didukung untuk kunci pribadi RSA. Selain itu, Anda tidak dapat menggunakan kunci pembungkus publik RSA_2048 dengan algoritma pembungkus RSAES_OAEP_SHA_* apa pun untuk membungkus kunci pribadi ECC_NIST_P521 (secp521r1). Anda harus menggunakan kunci pembungkus publik yang lebih besar atau algoritma pembungkus RSA_AES_KEY_WRAP.

Contoh berikut mengenkripsi materi kunci Anda dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_1, dan menyimpannya dalam file. `EncryptedKeyMaterial.bin`

Dalam contoh ini:

- *`WrappingPublicKey.bin`*adalah file yang berisi kunci publik pembungkus yang diunduh.
- *`PlaintextKeyMaterial.bin`*adalah file yang berisi materi kunci yang Anda enkripsi, seperti`PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` atau. `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 mendukung RSAES_OAEP_SHA_256 untuk kunci enkripsi simetris (SYMMETRIC_DEFAULT), kunci pribadi kurva elips (ECC), kunci pribadi SM2, dan kunci HMAC.

RSAES_OAEP_SHA_256 tidak didukung untuk kunci pribadi RSA. Selain itu, Anda tidak dapat menggunakan kunci pembungkus publik RSA_2048 dengan algoritma pembungkus RSAES_OAEP_SHA_* apa pun untuk membungkus kunci pribadi ECC_NIST_P521 (secp521r1). Anda harus menggunakan kunci publik yang lebih besar atau algoritma pembungkus RSA_AES_KEY_WRAP.

Contoh berikut mengenkripsi materi kunci dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_256, dan menyimpannya dalam file. `EncryptedKeyMaterial.bin`

Dalam contoh ini:

- *`WrappingPublicKey.bin`*adalah file yang berisi kunci pembungkus publik yang diunduh. Jika Anda mengunduh kunci publik dari konsol, file ini diberi nama `wrappingKey_`*`KMS`*

key_key_ID_timestamp (misalnya,wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909).

- *PlaintextKeyMaterial.bin*adalah file yang berisi materi kunci yang Anda enkripsi, seperti, PlaintextKeyMaterial.binHMAC_384_PlaintextKey.bin, atau. 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

Algoritma pembungkus RSA_AES_KEY_WRAP_SHA_1 melibatkan dua operasi enkripsi.

1. Enkripsi materi kunci Anda dengan kunci simetris AES yang Anda hasilkan dan algoritme enkripsi simetris AES.
2. Enkripsi kunci simetris AES yang Anda gunakan dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_1.

AWS KMS mendukung algoritma pembungkus RSA_AES_KEY_WRAP_SHA_* untuk semua jenis materi kunci impor yang didukung dan semua spesifikasi kunci publik yang didukung. Algoritma RSA_AES_KEY_WRAP_SHA_* adalah satu-satunya algoritma pembungkus yang didukung untuk membungkus materi kunci RSA.

Algoritma pembungkus RSA_AES_KEY_WRAP_SHA_1 membutuhkan OpenSSL versi 3. x atau yang lebih baru.

1. Hasilkan kunci enkripsi simetris AES 256-bit

   Perintah ini menghasilkan kunci enkripsi simetris AES yang terdiri dari 256 bit acak, dan menyimpannya dalam file aes-key.bin

```
# Generate a 32-byte AES symmetric encryption key
$ openssl rand -out aes-key.bin 32
```

2. Enkripsi materi kunci Anda dengan kunci enkripsi simetris AES

   Perintah ini mengenkripsi materi kunci Anda dengan kunci enkripsi simetris AES dan menyimpan materi kunci terenkripsi dalam file. `key-material-wrapped.bin`

   Dalam contoh perintah ini:

   - *`PlaintextKeyMaterial.bin`*adalah file yang berisi materi utama yang Anda impor, seperti,`PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.binRSA_3072_PrivateKey.der`, atau`ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`*adalah file yang berisi kunci enkripsi simetris AES 256-bit yang Anda buat di perintah sebelumnya.

```
# 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. Enkripsi kunci enkripsi simetris AES Anda dengan kunci publik

   Perintah ini mengenkripsi kunci enkripsi simetris AES Anda dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_1, Der menyandikannya, dan menyimpannya dalam file. `aes-key-wrapped.bin`

   Dalam contoh perintah ini:

   - *`WrappingPublicKey.bin`*adalah file yang berisi kunci pembungkus publik yang diunduh. Jika Anda mengunduh kunci publik dari konsol, file ini diberi nama `wrappingKey_`*`KMS key_key_ID_timestamp`* (misalnya, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`
   - *`aes-key.bin`*adalah file yang berisi kunci enkripsi simetris AES 256-bit yang Anda buat dalam perintah pertama dalam urutan contoh ini.

```
# 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. Hasilkan file yang akan diimpor

   Gabungkan file dengan materi kunci terenkripsi dan file dengan kunci AES terenkripsi. Simpan dalam `EncryptedKeyMaterial.bin` file, yang merupakan file yang akan Anda impor di fileLangkah 4: Impor material kunci.

   Dalam contoh perintah ini:

   - *`key-material-wrapped.bin`*adalah file yang berisi materi kunci terenkripsi Anda.
   - *`aes-key-wrapped.bin`*adalah file yang berisi kunci enkripsi AES terenkripsi.

```
# 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

Algoritma pembungkus RSA_AES_KEY_WRAP_SHA_256 melibatkan dua operasi enkripsi.

1. Enkripsi materi kunci Anda dengan kunci simetris AES yang Anda hasilkan dan algoritme enkripsi simetris AES.
2. Enkripsi kunci simetris AES yang Anda gunakan dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_256.

AWS KMS mendukung algoritma pembungkus RSA_AES_KEY_WRAP_SHA_* untuk semua jenis materi kunci impor yang didukung dan semua spesifikasi kunci publik yang didukung. Algoritma

RSA_AES_KEY_WRAP_SHA_* adalah satu-satunya algoritma pembungkus yang didukung untuk membungkus materi kunci RSA.

Algoritma pembungkus RSA_AES_KEY_WRAP_SHA_256 membutuhkan OpenSSL versi 3. x atau yang lebih baru.

1. Hasilkan kunci enkripsi simetris AES 256-bit

   Perintah ini menghasilkan kunci enkripsi simetris AES yang terdiri dari 256 bit acak, dan menyimpannya dalam file `aes-key.bin`

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. Enkripsi materi kunci Anda dengan kunci enkripsi simetris AES

   Perintah ini mengenkripsi materi kunci Anda dengan kunci enkripsi simetris AES dan menyimpan materi kunci terenkripsi dalam file. `key-material-wrapped.bin`

   Dalam contoh perintah ini:

   - *`PlaintextKeyMaterial.bin`*adalah file yang berisi materi utama yang Anda impor, seperti,`PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.binRSA_3072_PrivateKey.der`, atau`ECC_NIST_P521_PrivateKey.der`.
   - *`aes-key.bin`*adalah file yang berisi kunci enkripsi simetris AES 256-bit yang Anda buat di perintah sebelumnya.

   ```
   # 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. Enkripsi kunci enkripsi simetris AES Anda dengan kunci publik

   Perintah ini mengenkripsi kunci enkripsi simetris AES Anda dengan kunci publik yang Anda unduh dan algoritma pembungkus RSAES_OAEP_SHA_256, Der menyandikannya, dan menyimpannya dalam file. `aes-key-wrapped.bin`

Dalam contoh perintah ini:

- *WrappingPublicKey.bin*adalah file yang berisi kunci pembungkus publik yang diunduh. Jika Anda mengunduh kunci publik dari konsol, file ini diberi nama `wrappingKey_`*KMS key_key_ID_timestamp* (misalnya, `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`
- *aes-key.bin*adalah file yang berisi kunci enkripsi simetris AES 256-bit yang Anda buat dalam perintah pertama dalam urutan contoh ini.

```
# 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. Hasilkan file yang akan diimpor

   Gabungkan file dengan materi kunci terenkripsi dan file dengan kunci AES terenkripsi. Simpan dalam `EncryptedKeyMaterial.bin` file, yang merupakan file yang akan Anda impor di fileLangkah 4: Impor material kunci.

   Dalam contoh perintah ini:

   - *key-material-wrapped.bin*adalah file yang berisi materi kunci terenkripsi Anda.
   - *aes-key-wrapped.bin*adalah file yang berisi kunci enkripsi AES terenkripsi.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Lanjut ke Langkah 4: Impor material kunci.

Mengimpor material kunci langkah 4: Impor material kunci

Setelah Anda mengenkripsi materi kunci Anda, Anda dapat mengimpor materi kunci untuk digunakan dengan fileAWS KMS key. Untuk mengimpor material kunci, Anda mengunggah materi kunci terenkripsi dari Langkah 3: Enkripsi material kunci dan token impor yang Anda unduh di Langkah 2: Unduh kunci publik pembungkus dan token impor. Anda harus mengimpor materi kunci ke kunci KMS yang sama dengan yang Anda tentukan saat mengunduh kunci publik dan token impor. Ketika materi kunci berhasil diimpor, status kunci kunci KMS berubah menjadi`Enabled`, dan Anda dapat menggunakan kunci KMS dalam operasi kriptografi.

Saat Anda mengimpor materi kunci, Anda dapat mengatur waktu kedaluwarsa opsional untuk materi utama. Ketika materi kunci kedaluwarsa, AWS KMS menghapus materi kunci dan kunci KMS menjadi tidak dapat digunakan. Untuk menggunakan kunci KMS dalam operasi kriptografi, Anda harus mengimpor ulang materi kunci yang sama. Setelah mengimpor materi kunci, Anda tidak dapat mengatur, mengubah, atau membatalkan tanggal kedaluwarsa untuk impor saat ini. Untuk mengubah nilai-nilai ini, Anda harus menghapus dan mengimpor ulang materi kunci yang sama.

Untuk mengimpor materi kunci, Anda dapat menggunakan AWS KMS konsol atau ImportKeyMaterialAPI. Anda dapat menggunakan API secara langsung dengan membuat permintaan HTTP, atau dengan menggunakan AWSSDK, AWS Command Line Interfaceatau AWS Tools for PowerShell.

Saat Anda mengimpor materi kunci, ImportKeyMaterialentri ditambahkan ke AWS CloudTrail log Anda untuk merekam `ImportKeyMaterial` operasi. CloudTrail Entri adalah sama apakah Anda menggunakan AWS KMS konsol atau AWS KMS API.

Mengatur waktu kedaluwarsa (opsional)

Ketika Anda mengimpor materi kunci untuk kunci KMS Anda, Anda dapat menetapkan tanggal kedaluwarsa opsional dan waktu untuk materi kunci hingga 365 hari dari tanggal impor. Ketika materi kunci yang diimpor kedaluwarsa, AWS KMS hapus itu. Tindakan ini mengubah status kunci kunci KMS`PendingImport`, yang mencegahnya digunakan dalam operasi kriptografi apa pun. Untuk menggunakan kunci KMS, Anda harus mengimpor ulang salinan materi kunci asli.

Memastikan bahwa materi kunci impor sering kedaluwarsa dapat membantu Anda memenuhi persyaratan peraturan, tetapi menimbulkan risiko tambahan pada data yang dienkripsi di bawah kunci KMS. Sampai Anda mengimpor ulang salinan materi kunci asli, kunci KMS dengan materi kunci kedaluwarsa tidak dapat digunakan, dan data apa pun yang dienkripsi di bawah kunci KMS tidak dapat diakses. Jika Anda gagal mengimpor ulang materi kunci karena alasan apa pun, termasuk

kehilangan salinan materi kunci asli, kunci KMS tidak dapat digunakan secara permanen, dan data yang dienkripsi di bawah kunci KMS tidak dapat dipulihkan.

Untuk mengurangi risiko ini, pastikan salinan materi kunci impor Anda dapat diakses, dan rancang sistem untuk menghapus dan mengimpor kembali materi kunci sebelum kedaluwarsa dan mengganggu beban kerja Anda. AWS Kami menyarankan Anda menyetel alarm untuk kedaluwarsa materi kunci impor Anda yang memberi Anda banyak waktu untuk mengimpor kembali materi kunci sebelum kedaluwarsa. Anda juga dapat menggunakan CloudTrail log untuk mengaudit operasi yang mengimpor (dan mengimpor ulang) materi kunci dan menghapus materi kunci yang diimpor, dan AWS KMS operasi untuk menghapus materi kunci yang kedaluwarsa.

Anda tidak dapat mengimpor materi kunci yang berbeda ke dalam kunci KMS, dan AWS KMS tidak dapat memulihkan, memulihkan, atau mereproduksi materi kunci yang dihapus. Alih-alih menetapkan waktu kedaluwarsa, Anda dapat menghapus dan mengimpor ulang materi kunci yang diimpor secara terprogram secara berkala, tetapi persyaratan untuk menyimpan salinan materi kunci asli adalah sama.

Anda menentukan apakah dan kapan materi kunci impor kedaluwarsa saat Anda mengimpor materi kunci. Tetapi Anda dapat mengaktifkan dan menonaktifkan kedaluwarsa, atau mengatur waktu kedaluwarsa baru dengan menghapus dan mengimpor ulang materi kunci. Gunakan `ExpirationModel` parameter ImportKeyMaterialuntuk mengaktifkan expiration on (`KEY_MATERIAL_EXPIRES`) dan off (`KEY_MATERIAL_DOES_NOT_EXPIRE`) dan `ValidTo` parameter untuk mengatur waktu kedaluwarsa. Waktu maksimum adalah 365 hari dari data impor; tidak ada minimum, tetapi waktunya harus di masa depan.

Material kunci impor (konsol)

Anda dapat menggunakan AWS Management Console untuk mengimpor material kunci.

1. Jika Anda berada di halaman Unggah materi kunci yang dibungkus, lewati keStep 8.
2. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
3. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
4. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
5. Pilih ID kunci atau alias kunci KMS yang Anda unduh kunci publik dan token impor.
6. Pilih tab Konfigurasi kriptografi dan lihat nilainya. Tab ada di halaman detail untuk kunci KMS di bawah bagian konfigurasi Umum.

Anda hanya dapat mengimpor materi kunci ke kunci KMS dengan Asal Eksternal (Impor bahan kunci). Untuk informasi tentang membuat kunci KMS dengan materi kunci impor, lihatMengimpor bahan kunci untuk AWS KMS kunci.

7. Pilih tab Materi kunci dan kemudian pilih Impor bahan kunci. Tab Materi kunci hanya muncul untuk kunci KMS dengan nilai Asal Eksternal (Impor bahan kunci).

   Jika Anda mengunduh materi kunci, mengimpor token, dan mengenkripsi materi kunci, pilih Berikutnya.

8. Di bagian Materi kunci terenkripsi dan token impor, lakukan hal berikut.

   a. Di bawah Bahan kunci yang dibungkus, pilih Pilih file. Kemudian unggah file yang berisi material kunci Anda yang dibungkus (dienkripsi).

   b. Di bawah Impor token, pilih Pilih file. Unggah file yang berisi token impor yang Anda unduh.

9. Di bagian Opsi kedaluwarsa, Anda menentukan apakah material kunci kedaluwarsa. Untuk menetapkan tanggal dan waktu kedaluwarsa, pilih Material kunci kedaluwarsa, dan gunakan kalender untuk memilih tanggal dan waktu. Anda dapat menentukan tanggal hingga 365 hari dari tanggal dan waktu saat ini.

10. Pilih Unggah materi kunci.

Impor material kunci (API AWS KMS)

Untuk mengimpor bahan utama, gunakan ImportKeyMaterialoperasi. Contoh berikut menggunakan AWS CLI, tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Untuk menggunakan contoh ini:

1. Ganti *1234abcd-12ab-34cd-56ef-1234567890ab* dengan ID kunci kunci KMS yang Anda tentukan saat mengunduh kunci publik dan token impor. Untuk mengidentifikasi kunci KMS, gunakan ID kunci atau kunci ARN. Anda tidak dapat menggunakan nama alias atau alias ARN untuk operasi ini.
2. Ganti *EncryptedKeyMaterial.bin* dengan nama file yang berisi material kunci terenkripsi.
3. Ganti *ImportToken.bin* dengan nama file yang berisi token impor.
4. Jika Anda ingin materi kunci yang diimpor kedaluwarsa, atur nilai `expiration-model` parameter ke nilai defaultnya`KEY_MATERIAL_EXPIRES`, atau hilangkan parameternya. `expiration-model` Kemudian, ganti nilai `valid-to` parameter dengan tanggal dan waktu yang Anda inginkan materi kunci kedaluwarsa. Tanggal dan waktu bisa sampai 365 hari dari waktu permintaan.

```
$ 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
```

Jika Anda tidak ingin materi kunci yang diimpor kedaluwarsa, atur nilai `expiration-model` parameter ke `KEY_MATERIAL_DOES_NOT_EXPIRE` dan hilangkan `valid-to` parameter dari perintah.

```
$ 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**
>
> Jika perintah tidak berhasil, Anda mungkin melihat a `KMSInvalidStateException` atau a`NotFoundException`. Anda dapat mencoba kembali permintaan tersebut.

Penyimpanan kunci kustom

Penyimpanan kunci adalah lokasi yang aman untuk menyimpan kunci kriptografis. Penyimpanan kunci default AWS KMS juga mendukung metode untuk menghasilkan dan mengelola kunci yang disimpannya. Secara default, materi kunci kriptografi untuk AWS KMS keys yang Anda buat dihasilkan dan dilindungi oleh modul keamanan perangkat keras (HSM) yang AWS KMS merupakan modul kriptografi tervalidasi FIPS 140-2. Materi kunci untuk kunci KMS Anda tidak pernah meninggalkan HSM tidak terenkripsi.

Namun, jika Anda memerlukan lebih banyak kontrol dari HSM, Anda dapat membuat toko kunci khusus.

Toko kunci khusus adalah toko kunci logis di dalamnya AWS KMS yang didukung oleh manajer kunci di luar AWS KMS yang Anda miliki dan kelola. Toko kunci khusus menggabungkan antarmuka manajemen kunci yang nyaman dan komprehensif AWS KMS dengan kemampuan untuk memiliki dan mengontrol materi utama dan operasi kriptografi. Ketika Anda menggunakan kunci KMS di toko

kunci khusus, operasi kriptografi dilakukan oleh manajer kunci Anda menggunakan kunci kriptografi Anda. Akibatnya, Anda memikul lebih banyak tanggung jawab atas ketersediaan dan daya tahan kunci kriptografi, dan untuk pengoperasian HSM.

AWS KMSmendukung dua jenis toko kunci khusus.

- Toko AWS CloudHSM kunci adalah toko kunci AWS KMS khusus yang didukung oleh AWS CloudHSM cluster. Saat Anda membuat kunci KMS di toko kunci Anda, AWS KMS buat AWS CloudHSM kunci simetris Advanced Encryption Standard (AES) 256-bit, persisten, dan tidak dapat diekspor di klaster terkait. AWS CloudHSM Materi utama ini tidak pernah meninggalkan AWS CloudHSM cluster Anda tidak terenkripsi. Bila Anda menggunakan kunci KMS di AWS CloudHSM key store, operasi kriptografi dilakukan di HSM di cluster. AWS CloudHSMcluster didukung oleh modul keamanan perangkat keras (HSM) yang disertifikasi di FIPS 140-2 Level 3.
- Toko kunci eksternal adalah toko kunci AWS KMS khusus yang didukung oleh manajer kunci eksternal di luar AWS yang Anda miliki dan kendalikan. Ketika Anda menggunakan kunci KMS di toko kunci eksternal Anda, semua operasi enkripsi dan dekripsi dilakukan oleh manajer kunci eksternal Anda menggunakan kunci kriptografi Anda. Toko kunci eksternal dirancang untuk mendukung berbagai manajer kunci eksternal dari vendor yang berbeda.

  AWS KMSJangan pernah langsung melihat, mengakses, atau berinteraksi dengan manajer kunci eksternal atau kunci kriptografi Anda. Ketika Anda mengenkripsi atau mendekripsi dengan kunci KMS di toko kunci eksternal, operasi dilakukan oleh manajer kunci eksternal Anda menggunakan kunci eksternal Anda. Anda mempertahankan kendali penuh atas kunci kriptografi Anda, termasuk kemampuan untuk menolak atau menghentikan operasi kriptografi tanpa berinteraksi dengannya. AWS Namun, karena jarak dan pemrosesan ekstra, kunci KMS di toko kunci eksternal mungkin memiliki latensi dan kinerja yang lebih buruk, dan mungkin memiliki karakteristik ketersediaan yang berbeda dari kunci KMS dengan bahan utama. AWS KMS Untuk informasi selengkapnya tentang pengelola kunci yang kompatibel dengan fitur penyimpanan kunci AWS KMS eksternal, lihat Vendor eksternal mana yang mendukung spesifikasi Proxy XKS? di AWS Key Management ServiceFAQ.

Kedua jenis toko kunci khusus ini sangat berbeda dari toko AWS KMS kunci standar dan satu sama lain. Model keamanan mereka, fokus tanggung jawab, kinerja, harga, dan kasus penggunaan juga sangat berbeda. Sebelum memilih toko kunci khusus, baca dokumentasi terkait dan konfirmasikan bahwa konfigurasi tambahan dan tanggung jawab pemeliharaan adalah pertukaran yang bijaksana untuk kontrol ekstra. Namun, jika aturan dan peraturan di mana Anda beroperasi memerlukan kontrol langsung atas materi utama, toko kunci khusus mungkin merupakan pilihan yang baik untuk Anda.

Fitur yang tidak didukung

AWS KMStidak mendukung fitur berikut di toko kunci khusus.

- Kunci Asymmetric KMS
- Pasangan kunci data asimetris
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor
- Rotasi kunci otomatis
- Kunci Multi-Region

Topik

- AWS CloudHSM toko-toko utama
- Toko kunci eksternal

AWS CloudHSM toko-toko utama

Toko AWS CloudHSM kunci adalah toko kunci khusus yang didukung oleh AWS CloudHSM cluster. Saat Anda membuat AWS KMS keydi penyimpanan kunci khusus, buat dan simpan materi AWS KMS kunci yang tidak dapat diekstraksi untuk kunci KMS dalam AWS CloudHSM klaster yang Anda miliki dan kelola. Bila Anda menggunakan kunci KMS di toko kunci kustom, operasi kriptografi dilakukan di HSM di cluster. Fitur ini menggabungkan kenyamanan dan integrasi luas AWS KMS dengan kontrol tambahan AWS CloudHSM cluster di Anda Akun AWS.

AWS KMS menyediakan konsol lengkap dan dukungan API untuk membuat, menggunakan, dan mengelola toko kunci kustom Anda. Anda dapat menggunakan kunci KMS di toko kunci kustom Anda dengan cara yang sama seperti Anda menggunakan kunci KMS apa pun. Misalnya, Anda dapat menggunakan kunci KMS untuk menghasilkan kunci data dan mengenkripsi data. Anda juga dapat menggunakan kunci KMS di toko kunci kustom Anda dengan AWS layanan yang mendukung kunci yang dikelola pelanggan.

Apakah saya memerlukan toko kunci khusus?

Bagi sebagian besar pengguna, penyimpanan AWS KMS kunci default, yang dilindungi oleh modul kriptografi tervalidasi FIPS 140-2, memenuhi persyaratan keamanan mereka. Tidak perlu menambahkan lapisan tambahan dari tanggung jawab pemeliharaan atau dependensi pada layanan tambahan.

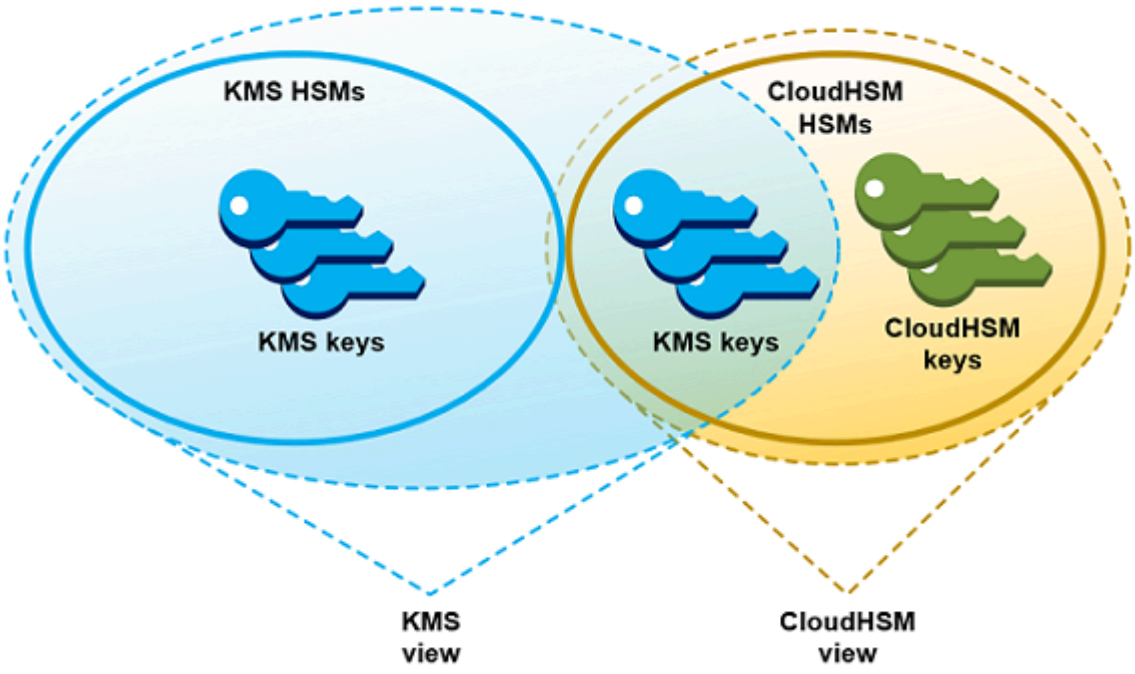
Namun, Anda dapat mempertimbangkan untuk membuat penyimpanan kunci kustom jika organisasi Anda memiliki salah satu persyaratan berikut:

- Anda memiliki kunci yang secara eksplisit diperlukan untuk dilindungi dalam HSM penyewa tunggal atau dalam HSM yang Anda memiliki kendali langsung atas.
- Anda membutuhkan kemampuan untuk segera menghapus materi kunci dari AWS KMS.
- Anda harus dapat mengaudit semua penggunaan kunci Anda secara independen dari AWS KMS atau AWS CloudTrail.

Bagaimana cara kerja toko kunci khusus?

Setiap toko kunci khusus dikaitkan dengan AWS CloudHSM cluster di Anda Akun AWS. Saat Anda menghubungkan penyimpanan kunci khusus ke klasternya, AWS KMS buat infrastruktur jaringan untuk mendukung koneksi. Kemudian masuk ke AWS CloudHSM klien kunci di cluster menggunakan kredensyal pengguna crypto khusus di cluster.

Anda membuat dan mengelola toko kunci kustom Anda di AWS KMS dan membuat dan mengelola klaster HSM Anda di. AWS CloudHSM Saat Anda membuat AWS KMS keys di toko kunci AWS KMS khusus, Anda melihat dan mengelola kunci KMS di AWS KMS. Tetapi Anda juga dapat melihat dan mengelola materi kunci mereka AWS CloudHSM, seperti yang akan Anda lakukan untuk kunci lain di cluster.



Anda dapat membuat kunci KMS enkripsi simetris dengan materi kunci yang dihasilkan oleh AWS KMS di toko kunci kustom Anda. Kemudian gunakan teknik yang sama untuk melihat dan mengelola kunci KMS di toko kunci kustom Anda yang Anda gunakan untuk kunci KMS di toko AWS KMS kunci. Anda dapat mengontrol akses dengan IAM dan kebijakan kunci, membuat tag dan alias, mengaktifkan dan menonaktifkan kunci KMS, dan menjadwalkan penghapusan kunci. Anda dapat menggunakan kunci KMS untuk operasi kriptografi dan menggunakannya dengan AWS layanan yang terintegrasi dengannya. AWS KMS

Selain itu, Anda memiliki kontrol penuh atas AWS CloudHSM cluster, termasuk membuat dan menghapus HSM dan mengelola cadangan. Anda dapat menggunakan AWS CloudHSM klien dan pustaka perangkat lunak yang didukung untuk melihat, mengaudit, dan mengelola materi utama untuk kunci KMS Anda. Sementara toko kunci khusus terputus, AWS KMS tidak dapat mengaksesnya, dan pengguna tidak dapat menggunakan kunci KMS di toko kunci khusus untuk operasi kriptografi. Lapisan kontrol tambahan ini membuat penyimpanan kunci kustom menjadi solusi yang ampuh untuk organisasi yang memerlukannya.

Di mana saya mulai?

Untuk membuat dan mengelola toko AWS CloudHSM kunci, Anda menggunakan fitur AWS KMS dan AWS CloudHSM.

1. Mulai masuk AWS CloudHSM. Buat klaster AWS CloudHSM aktif atau pilih klaster yang ada. Klaster harus memiliki setidaknya dua HSM aktif di Availability Zone yang berbeda. Kemudian buat akun pengguna kripto (CU) khusus di klaster tersebut untuk AWS KMS.
2. Di AWS KMS, buat penyimpanan kunci khusus yang terkait dengan AWS CloudHSM cluster yang Anda pilih. AWS KMS menyediakan antarmuka manajemen lengkap yang memungkinkan Anda membuat, melihat, mengedit, dan menghapus toko kunci kustom Anda.
3. Saat Anda siap menggunakan toko kunci kustom Anda, sambungkan ke AWS CloudHSM klaster terkait. AWS KMS menciptakan infrastruktur jaringan yang dibutuhkan untuk mendukung koneksi. Ini kemudian login ke klaster menggunakan kredensial akun pengguna kripto khusus sehingga dapat menghasilkan dan mengelola material kunci dalam klaster.
4. Sekarang, Anda dapat membuat kunci KMS enkripsi simetris di toko kunci khusus Anda. Cukup tentukan toko kunci khusus saat Anda membuat kunci KMS.

Jika Anda bingung pada suatu titik, Anda dapat menemukan bantuan di topik Memecahkan masalah penyimpanan kunci kustom. Jika pertanyaan Anda tidak terjawab, gunakan tautan umpan balik di

bagian bawah setiap halaman panduan ini atau posting pertanyaan di Forum Diskusi AWS Key Management Service.

Kuota

AWS KMS memungkinkan hingga 10 toko kunci khusus di masing-masing Akun AWS dan Wilayah, termasuk toko utama dan toko AWS CloudHSM kunci eksternal, terlepas dari status koneksi mereka. Selain itu, ada kuota AWS KMS permintaan tentang penggunaan kunci KMS di toko AWS CloudHSM kunci.

Penetapan Harga

Untuk informasi tentang biaya toko kunci AWS KMS khusus dan kunci yang dikelola pelanggan di toko kunci kustom, lihat AWS Key Management Service harga. Untuk informasi tentang biaya AWS CloudHSM cluster dan HSM, lihat AWS CloudHSM Harga.

Daerah

AWS KMS mendukung toko-toko AWS CloudHSM utama di semua Wilayah AWS tempat yang AWS KMS didukung, kecuali untuk Asia Pasifik (Melbourne), China (Beijing), China (Ningxia), dan Eropa (Spanyol).

Fitur yang tidak didukung

AWS KMS tidak mendukung fitur berikut di toko kunci khusus.

- Kunci Asymmetric KMS
- Pasangan kunci data asimetris
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor
- Rotasi kunci otomatis
- Kunci Multi-Region

Topik

- AWS CloudHSMkonsep toko utama
- Mengontrol akses ke toko AWS CloudHSM utama Anda
- Mengelola toko kunci kustom CloudHSM

- Mengelola kunci KMS di toko kunci CloudHSM
- Memecahkan masalah penyimpanan kunci kustom

AWS CloudHSMkonsep toko utama

Topik ini menjelaskan beberapa konsep yang digunakan di toko-toko AWS CloudHSM utama.

AWS CloudHSMtoko kunci

Toko AWS CloudHSM kunci adalah toko kunci khusus yang terkait dengan AWS CloudHSM cluster yang Anda miliki dan kelola. AWS CloudHSMcluster didukung oleh modul keamanan perangkat keras (HSM) yang disertifikasi di FIPS 140-2 Level 3.

Saat Anda membuat kunci KMS di toko kunci Anda, AWS KMS buat AWS CloudHSM kunci simetris Advanced Encryption Standard (AES) 256-bit, persisten, dan tidak dapat diekspor di klaster terkait. AWS CloudHSM Material utama ini tidak pernah meninggalkan HSM Anda tidak terenkripsi. Ketika Anda menggunakan kunci KMS di toko AWS CloudHSM kunci, operasi kriptografi dilakukan di HSM di cluster.

AWS CloudHSMtoko utama menggabungkan antarmuka manajemen kunci yang nyaman dan komprehensif AWS KMS dengan kontrol tambahan yang disediakan oleh AWS CloudHSM cluster di AndaAkun AWS. Fitur terintegrasi ini memungkinkan Anda membuat, mengelola, dan menggunakan kunci KMS AWS KMS sambil mempertahankan kontrol penuh dari HSM yang menyimpan materi utama mereka, termasuk mengelola cluster, HSM, dan backup. Anda dapat menggunakan AWS KMS konsol dan API untuk mengelola penyimpanan AWS CloudHSM kunci dan kunci KMS-nya. Anda juga dapat menggunakan konsol AWS CloudHSM, API, perangkat lunak klien, dan pustaka perangkat lunak terkait untuk mengelola klaster terkait.

Anda dapat melihat dan mengelola toko AWS CloudHSM kunci Anda, mengedit propertinya, dan menghubungkan dan memutusnya dari AWS CloudHSM klaster terkait. Jika Anda perlu menghapus toko AWS CloudHSM kunci, Anda harus terlebih dahulu menghapus kunci KMS di toko AWS CloudHSM kunci dengan menjadwalkan penghapusan mereka dan menunggu sampai masa tenggang berakhir. Menghapus penyimpanan AWS CloudHSM kunci menghapus sumber daya dariAWS KMS, tetapi itu tidak mempengaruhi AWS CloudHSM cluster Anda.

Klaster AWS CloudHSM

Setiap toko AWS CloudHSM kunci dikaitkan dengan satu AWS CloudHSM cluster. Saat Anda membuat AWS KMS key di toko AWS CloudHSM kunci Anda, AWS KMS buat materi kuncinya di

cluster terkait. Ketika Anda menggunakan kunci KMS di toko AWS CloudHSM kunci Anda, operasi kriptografi dilakukan di cluster terkait.

Setiap AWS CloudHSM cluster dapat dikaitkan dengan hanya satu penyimpanan AWS CloudHSM kunci. Cluster yang Anda pilih tidak dapat dikaitkan dengan penyimpanan AWS CloudHSM kunci lain atau berbagi riwayat cadangan dengan klaster yang terkait dengan penyimpanan AWS CloudHSM kunci lain. Cluster harus diinisialisasi dan aktif, dan harus sama Akun AWS dan Region sebagai penyimpanan AWS CloudHSM kunci. Anda dapat membuat klaster baru atau menggunakan yang sudah ada. AWS KMS tidak memerlukan penggunaan klaster eksklusif. Untuk membuat kunci KMS di toko AWS CloudHSM kunci, cluster terkait itu harus berisi setidaknya dua HSM aktif. Semua operasi lain hanya memerlukan satu HSM.

Anda menentukan AWS CloudHSM cluster saat Anda membuat penyimpanan AWS CloudHSM kunci, dan Anda tidak dapat mengubahnya. Namun, Anda dapat mengganti setiap klaster yang berbagi riwayat cadangan dengan klaster asli. Hal ini memungkinkan Anda menghapus klaster, jika perlu, dan menggantinya dengan klaster yang dibuat dari salah satu cadangan. Anda mempertahankan kontrol penuh dari klaster AWS CloudHSM yang terkait, sehingga Anda dapat mengelola pengguna dan kunci, membuat dan menghapus HSM, serta menggunakan dan mengelola cadangan.

Ketika Anda siap untuk menggunakan toko AWS CloudHSM kunci Anda, Anda menghubungkannya ke AWS CloudHSM cluster terkait. Anda dapat menghubungkan dan memutus penyimpanan kunci kustom Anda kapan saja. Ketika toko kunci khusus terhubung, Anda dapat membuat dan menggunakan kunci KMS-nya. Ketika terputus, Anda dapat melihat dan mengelola toko AWS CloudHSM kunci dan kunci KMS-nya. Tetapi Anda tidak dapat membuat kunci KMS baru atau menggunakan kunci KMS di toko AWS CloudHSM kunci untuk operasi kriptografi.

Pengguna kripto **`kmsuser`**

Untuk membuat dan mengelola material kunci dalam klaster AWS CloudHSM terkait atas nama Anda, AWS KMS menggunakan pengguna kripto (CU) AWS CloudHSM khusus dalam klaster bernama `kmsuser`. CU `kmsuser` adalah akun CU standar yang secara otomatis disinkronkan untuk semua HSM di klaster dan disimpan dalam cadangan klaster.

Sebelum Anda membuat penyimpanan AWS CloudHSM kunci, Anda membuat akun `kmsuser` CU di AWS CloudHSM cluster Anda menggunakan perintah createUser di cloudhsm_mgmt_util. Kemudian ketika Anda membuat toko AWS CloudHSM kunci, Anda memberikan kata sandi `kmsuser` akun keAWS KMS. Saat Anda menghubungkan toko kunci khusus, AWS KMS masuk ke cluster sebagai `kmsuser` CU dan memutar kata sandinya. AWS KMSmengenkripsi `kmsuser` kata sandi Anda

sebelum menyimpannya dengan aman. Ketika kata sandi diputar, kata sandi baru dienkripsi dan disimpan dengan cara yang sama.

AWS KMStetap masuk `kmsuser` selama toko AWS CloudHSM kunci terhubung. Anda tidak boleh menggunakan akun CU ini untuk tujuan lain. Namun, Anda mempertahankan kontrol tertinggi dari akun CU `kmsuser`. Kapan saja, Anda dapat menemukan handel kunci dari kunci yang dimiliki oleh `kmsuser`. Jika perlu, Anda dapat memutuskan koneksi penyimpanan kunci kustom, mengubah kata sandi `kmsuser`,login ke klaster sebagai `kmsuser`, serta melihat dan mengelola kunci yang dimiliki `kmsuser`.

Untuk petunjuk tentang cara membuat akun CU `kmsuser` Anda, lihat Membuat Pengguna Kripto `kmsuser`.

Kunci KMS di toko AWS CloudHSM kunci

Anda dapat menggunakan AWS KMS API AWS KMS atau untuk membuat AWS KMS keysdi toko AWS CloudHSM kunci. Anda menggunakan teknik yang sama yang akan Anda gunakan pada kunci KMS apa pun. Satu-satunya perbedaan adalah Anda harus mengidentifikasi penyimpanan AWS CloudHSM kunci dan menentukan bahwa asal bahan utama adalah AWS CloudHSM cluster.

Saat Anda membuat kunci KMS di penyimpanan kunci, buat AWS CloudHSM kunci KMS AWS KMS dan itu menghasilkan AWS KMS materi kunci simetris Advanced Encryption Standard (AES) 256-bit, persisten, dan tidak dapat diekspor di klaster terkaitnya. Ketika Anda menggunakan AWS KMS kunci dalam operasi kriptografi, operasi dilakukan di AWS CloudHSM cluster menggunakan kunci AES berbasis cluster. Meskipun AWS CloudHSM mendukung kunci simetris dan asimetris dari berbagai jenis, toko AWS CloudHSM kunci hanya mendukung kunci enkripsi simetris AES.

Anda dapat melihat kunci KMS di penyimpanan AWS CloudHSM kunci di AWS KMS konsol, dan menggunakan opsi konsol untuk menampilkan ID penyimpanan kunci khusus. Anda juga dapat menggunakan DescribeKeyoperasi untuk menemukan ID penyimpanan AWS CloudHSM kunci dan ID AWS CloudHSM cluster.

Kunci KMS di toko AWS CloudHSM kunci berfungsi seperti kunci KMS apa pun. AWS KMS Pengguna yang berwenang memerlukan izin yang sama untuk menggunakan dan mengelola kunci KMS. Anda menggunakan prosedur konsol dan operasi API yang sama untuk melihat dan mengelola kunci KMS di penyimpanan AWS CloudHSM kunci. Ini termasuk mengaktifkan dan menonaktifkan kunci KMS, membuat dan menggunakan tag dan alias, dan mengatur dan mengubah IAM dan kebijakan utama. Anda dapat menggunakan kunci KMS di toko AWS CloudHSM kunci untuk operasi kriptografi, dan menggunakannya dengan AWSlayanan terintegrasi yang mendukung penggunaan

kunci yang dikelola pelanggan Namun, Anda tidak dapat mengaktifkan rotasi kunci otomatis atau mengimpor bahan kunci ke kunci KMS di toko kunci. AWS CloudHSM

Anda juga menggunakan proses yang sama untuk menjadwalkan penghapusan kunci KMS di toko kunci. AWS CloudHSM Setelah masa tunggu berakhir, AWS KMS hapus kunci KMS dari KMS. Kemudian itu membuat upaya terbaik untuk menghapus materi kunci untuk kunci KMS dari AWS CloudHSM cluster terkait. Namun, Anda mungkin perlu secara manual menghapus material kunci tanpa induk dari klaster dan cadangannya.

Mengontrol akses ke toko AWS CloudHSM utama Anda

Anda menggunakan kebijakan IAM untuk mengontrol akses ke penyimpanan AWS CloudHSM kunci dan AWS CloudHSM klaster Anda. Anda dapat menggunakan kebijakan utama, kebijakan IAM, dan hibah untuk mengontrol akses ke toko AWS KMS keys AWS CloudHSM kunci Anda. Sebaiknya Anda menyediakan pengguna, grup, dan peran izin yang hanya mereka perlukan untuk tugas-tugas yang sangat mungkin akan mereka lakukan.

Topik

- Mengotorisasi manajer dan pengguna toko AWS CloudHSM utama
- Mengizinkan AWS KMS mengelola sumber daya AWS CloudHSM Amazon EC2

Mengotorisasi manajer dan pengguna toko AWS CloudHSM utama

Saat mendesain toko AWS CloudHSM kunci Anda, pastikan bahwa kepala sekolah yang menggunakan dan mengelolanya hanya memiliki izin yang mereka butuhkan. Daftar berikut menjelaskan izin minimum yang diperlukan untuk pengelola dan pengguna toko AWS CloudHSM utama.

- Prinsipal yang membuat dan mengelola toko AWS CloudHSM kunci Anda memerlukan izin berikut untuk menggunakan operasi API penyimpanan AWS CloudHSM kunci.
  - `cloudhsm:DescribeClusters`
  - `kms:CreateCustomKeyStore`
  - `kms:ConnectCustomKeyStore`
  - `kms:DeleteCustomKeyStore`
  - `kms:DescribeCustomKeyStores`
  - `kms:DisconnectCustomKeyStore`
  - `kms:UpdateCustomKeyStore`

  - `iam:CreateServiceLinkedRole`
- Prinsipal yang membuat dan mengelola AWS CloudHSM klaster yang terkait dengan penyimpanan AWS CloudHSM kunci Anda memerlukan izin untuk membuat dan menginisialisasi cluster. AWS CloudHSM Ini termasuk izin untuk membuat atau menggunakan Amazon Virtual Private Cloud (VPC), membuat subnet, dan membuat instans Amazon EC2. Mereka mungkin juga perlu membuat dan menghapus HSM, serta mengelola cadangan. Untuk daftar izin yang diperlukan, lihat Identitas dan manajemen akses AWS CloudHSM di Panduan AWS CloudHSM Pengguna.
- Prinsipal yang membuat dan mengelola AWS KMS keys di toko AWS CloudHSM kunci Anda memerlukan izin yang sama dengan mereka yang membuat dan mengelola kunci KMS apa pun. AWS KMS Kebijakan kunci default untuk kunci KMS di penyimpanan AWS CloudHSM kunci identik dengan kebijakan kunci default untuk kunci KMS di. AWS KMS Attribute-based access control (ABAC), yang menggunakan tag dan alias untuk mengontrol akses ke kunci KMS, juga efektif pada kunci KMS di toko-toko utama. AWS CloudHSM
- Prinsipal yang menggunakan kunci KMS di toko AWS CloudHSM kunci Anda untuk operasi kriptografi memerlukan izin untuk melakukan operasi kriptografi dengan kunci KMS, seperti KMS: Decrypt. Anda dapat memberikan izin ini dalam kebijakan utama, kebijakan IAM. Tapi, mereka tidak memerlukan izin tambahan untuk menggunakan kunci KMS di toko AWS CloudHSM kunci.

Mengizinkan AWS KMS mengelola sumber daya AWS CloudHSM Amazon EC2

Untuk mendukung toko AWS CloudHSM utama Anda, AWS KMS perlu izin untuk mendapatkan informasi tentang AWS CloudHSM cluster Anda. Ini juga membutuhkan izin untuk membuat infrastruktur jaringan yang menghubungkan toko AWS CloudHSM kunci Anda ke AWS CloudHSM klasternya. Untuk mendapatkan izin ini, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan di Anda. Akun AWS Pengguna yang membuat toko AWS CloudHSM kunci harus memiliki `iam:CreateServiceLinkedRole` izin yang memungkinkan mereka membuat peran terkait layanan.

Topik

- Tentang peran AWS KMS yang tertaut layanan
- Membuat peran yang ditautkan ke layanan
- Mengedit deskripsi peran tertaut layanan
- Menghapus peran tertaut layanan

Tentang peran AWS KMS yang tertaut layanan

Peran yang tertaut dengan layanan adalah IAM role yang memberikan satu izin layanan AWS untuk memanggil layanan AWS lainnya atas nama Anda. Ini dirancang agar mempermudah Anda menggunakan fitur dari beberapa layanan AWS terintegrasi tanpa harus membuat dan memelihara kebijakan IAM yang kompleks. Untuk informasi selengkapnya, lihat Menggunakan peran tertaut layanan untuk AWS KMS.

Untuk penyimpanan AWS CloudHSM utama, AWS KMS buat peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan dengan kebijakan. AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy Kebijakan ini memberi peran izin berikut:

- Cloudhsm:describe* — mendeteksi perubahan dalam AWS CloudHSM cluster yang dilampirkan ke toko kunci kustom Anda.
- ec2: CreateSecurityGroup — digunakan saat Anda menghubungkan toko AWS CloudHSM kunci untuk membuat grup keamanan yang memungkinkan arus lalu lintas jaringan antara AWS KMS dan AWS CloudHSM cluster Anda.
- ec2: AuthorizeSecurityGroupIngress — digunakan saat Anda menghubungkan toko AWS CloudHSM kunci untuk memungkinkan akses jaringan dari AWS KMS ke VPC yang berisi AWS CloudHSM cluster Anda.
- ec2: CreateNetworkInterface — digunakan ketika Anda menghubungkan toko AWS CloudHSM kunci untuk membuat antarmuka jaringan yang digunakan untuk komunikasi antara AWS KMS dan AWS CloudHSM cluster.
- ec2: RevokeSecurityGroupEgress — digunakan saat Anda menghubungkan toko AWS CloudHSM kunci untuk menghapus semua aturan keluar dari grup keamanan yang AWS KMS dibuat.
- ec2: DeleteSecurityGroup — digunakan saat Anda memutuskan penyimpanan AWS CloudHSM kunci untuk menghapus grup keamanan yang dibuat saat Anda menghubungkan toko AWS CloudHSM kunci.
- ec2: DescribeSecurityGroups — digunakan untuk memantau perubahan dalam grup keamanan yang AWS KMS dibuat di VPC yang berisi cluster AWS CloudHSM Anda sehingga dapat memberikan pesan kesalahan AWS KMS yang jelas jika terjadi kegagalan.
- ec2: DescribeVpcs — digunakan untuk memantau perubahan dalam VPC yang berisi cluster AWS CloudHSM Anda sehingga dapat memberikan pesan kesalahan AWS KMS yang jelas jika terjadi kegagalan.

- ec2: DescribeNetworkAcls — digunakan untuk memantau perubahan dalam ACL jaringan untuk VPC yang berisi AWS CloudHSM cluster Anda sehingga AWS KMS dapat memberikan pesan kesalahan yang jelas jika terjadi kegagalan.
- ec2: DescribeNetworkInterfaces — digunakan untuk memantau perubahan pada antarmuka jaringan yang AWS KMS dibuat di VPC yang berisi AWS CloudHSM cluster Anda sehingga AWS KMS dapat memberikan pesan kesalahan yang jelas jika terjadi kegagalan.

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

Karena peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan hanya mempercayai`cks.kms.amazonaws.com`, hanya AWS KMS dapat mengambil peran terkait layanan ini. Peran ini terbatas pada operasi yang AWS KMS perlu melihat AWS CloudHSM kluster Anda dan untuk menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klaster terkait. Ini tidak memberi AWS KMS izin tambahan apa pun. Misalnya, AWS KMS tidak memiliki izin untuk membuat, mengelola, atau menghapus perangkat klaster, HSM, atau pencadangan AWS CloudHSM.

Daerah

Seperti fitur toko AWS CloudHSM utama, AWSServiceRoleForKeyManagementServiceCustomKeyStoresperan ini didukung di semua Wilayah

AWS tempat AWS KMS dan AWS CloudHSM tersedia. Untuk daftar yang didukung Wilayah AWS oleh setiap layanan, lihat AWS Key Management ServiceTitik Akhir dan Kuota dan AWS CloudHSM titik akhir dan kuota di. Referensi Umum Amazon Web Services

Untuk informasi selengkapnya tentang cara AWS layanan menggunakan peran terkait layanan, lihat Menggunakan peran terkait layanan di Panduan Pengguna IAM.

Membuat peran yang ditautkan ke layanan

AWS KMSsecara otomatis membuat peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan di Akun AWS saat Anda membuat penyimpanan AWS CloudHSM kunci, jika peran tersebut belum ada. Anda tidak dapat membuat atau membuat ulang peran yang tertaut dengan layanan ini secara langsung.

Mengedit deskripsi peran tertaut layanan

Anda tidak dapat mengedit nama peran atau pernyataan kebijakan dalam peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan, tetapi Anda dapat mengedit deskripsi peran. Untuk petunjuknya, lihat Mengedit peran terkait layanan di Panduan Pengguna IAM.

Menghapus peran tertaut layanan

AWS KMStidak menghapus peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan dari Anda Akun AWS bahkan jika Anda telah menghapus semua toko AWS CloudHSM utama Anda. Meskipun saat ini tidak ada prosedur untuk menghapus peran AWSServiceRoleForKeyManagementServiceCustomKeyStoresterkait layanan, AWS KMS tidak mengambil peran ini atau menggunakan izinnya kecuali Anda memiliki penyimpanan kunci aktif. AWS CloudHSM

Mengelola toko kunci kustom CloudHSM

Dengan menggunakan AWS Management Console dan AWS KMSAPI, Anda dapat mengelola penyimpanan kunci kustom. Misalnya, Anda dapat melihat penyimpanan kunci kustom, mengedit propertinya, menghubungkan dan memutus koneksinya dari klaster AWS CloudHSM yang terkait, dan menghapus penyimpanan kunci kustom.

Topik

- Membuat toko AWS CloudHSM kunci
- Melihat toko AWS CloudHSM kunci
- Mengedit pengaturan toko AWS CloudHSM kunci
- Menghubungkan dan memutuskan sambungan toko AWS CloudHSM kunci
- Menghapus toko AWS CloudHSM kunci

Membuat toko AWS CloudHSM kunci

Anda dapat membuat satu atau beberapa toko AWS CloudHSM utama di akun Anda. Setiap toko AWS CloudHSM kunci dikaitkan dengan satu AWS CloudHSM cluster di wilayah Akun AWS dan yang sama. Sebelum Anda membuat toko AWS CloudHSM kunci Anda, Anda perlu merakit prasyarat. Kemudian, sebelum Anda dapat menggunakan toko AWS CloudHSM kunci Anda, Anda harus menghubungkannya ke AWS CloudHSM klasternya.

> **Note**
>
> Jika Anda mencoba membuat penyimpanan AWS CloudHSM kunci dengan semua nilai properti yang sama dengan penyimpanan AWS CloudHSM kunci terputus yang ada, AWS KMS tidak membuat penyimpanan AWS CloudHSM kunci baru, dan itu tidak memunculkan pengecualian atau menampilkan kesalahan. Sebaliknya, AWS KMS mengenali duplikat sebagai konsekuensi yang mungkin dari percobaan ulang, dan mengembalikan ID dari penyimpanan kunci yang adaAWS CloudHSM.

> **Tip**
>
> Anda tidak harus segera menghubungkan toko AWS CloudHSM kunci Anda. Anda dapat membiarkannya dalam keadaan terputus sampai Anda siap menggunakannya. Namun, untuk memverifikasi bahwa itu dikonfigurasi dengan benar, Anda mungkin ingin menghubungkannya, melihat status koneksinya, dan kemudian memutuskannya.

Topik

- Memasang prasyarat
- Buat toko AWS CloudHSM kunci (konsol)

- Buat toko AWS CloudHSM kunci (API)

Memasang prasyarat

Setiap toko AWS CloudHSM kunci didukung oleh sebuah AWS CloudHSM cluster. Untuk membuat penyimpanan AWS CloudHSM kunci, Anda harus menentukan AWS CloudHSM cluster aktif yang belum dikaitkan dengan penyimpanan kunci lain. Anda juga perlu membuat pengguna kripto (CU) khusus di HSM klaster yang dapat digunakan AWS KMS untuk membuat dan mengelola kunci atas nama Anda.

Sebelum Anda membuat toko AWS CloudHSM kunci, lakukan hal berikut:

Pilih klaster AWS CloudHSM

Setiap toko AWS CloudHSM kunci dikaitkan dengan tepat satu AWS CloudHSM cluster. Saat Anda membuat AWS KMS keysdi penyimpanan AWS CloudHSM kunci, AWS KMS buat metadata kunci KMS, seperti ID dan Nama Sumber Daya Amazon (ARN) di. AWS KMS Ini kemudian membuat materi kunci dalam HSM dari klaster terkait. Anda dapat membuat klaster AWS CloudHSM baru atau menggunakan yang sudah ada. AWS KMS tidak memerlukan akses eksklusif ke klaster.

AWS CloudHSMCluster yang Anda pilih secara permanen terkait dengan penyimpanan AWS CloudHSM kunci. Setelah Anda membuat penyimpanan AWS CloudHSM kunci, Anda dapat mengubah ID cluster dari cluster terkait, tetapi cluster yang Anda tentukan harus berbagi riwayat cadangan dengan cluster asli. Untuk menggunakan cluster yang tidak terkait, Anda perlu membuat toko AWS CloudHSM kunci baru.

Klaster AWS CloudHSM yang Anda pilih harus memiliki karakteristik sebagai berikut:

- Klaster harus aktif.

  Anda harus membuat klaster, menginisialisasinya, menginstal perangkat lunak klien AWS CloudHSM untuk platform Anda, kemudian mengaktifkan klaster. Untuk petunjuk terperinci, lihat Memulai AWS CloudHSM di Panduan AWS CloudHSM Pengguna.
- Cluster harus berada di akun dan Wilayah yang sama dengan penyimpanan AWS CloudHSM kunci. Anda tidak dapat mengaitkan penyimpanan AWS CloudHSM kunci di satu Wilayah dengan klaster di Wilayah yang berbeda. Untuk membuat infrastruktur utama di beberapa Wilayah, Anda harus membuat penyimpanan dan klaster AWS CloudHSM utama di setiap Wilayah.

- Cluster tidak dapat dikaitkan dengan penyimpanan kunci kustom lain di akun dan Wilayah yang sama. Setiap penyimpanan AWS CloudHSM kunci di akun dan Wilayah harus dikaitkan dengan AWS CloudHSM cluster yang berbeda. Anda tidak dapat menentukan klaster yang sudah dikaitkan dengan penyimpanan kunci kustom atau klaster yang berbagi riwayat pencadangan dengan klaster yang dikaitkan. Klaster yang berbagi riwayat pencadangan memiliki sertifikat klaster yang sama. Untuk melihat sertifikat cluster cluster, gunakan AWS CloudHSM konsol atau DescribeClustersoperasi.

  Jika Anda mencadangkan AWS CloudHSM klaster ke Wilayah yang berbeda, itu dianggap sebagai klaster yang berbeda, dan Anda dapat mengaitkan cadangan dengan penyimpanan kunci khusus di Wilayahnya. Namun, kunci KMS di dua toko kunci khusus tidak dapat dioperasikan, bahkan jika mereka memiliki kunci dukungan yang sama. AWS KMSmengikat metadata ke ciphertext sehingga dapat didekripsi hanya dengan kunci KMS yang mengenkripsi itu.

- Klaster harus dikonfigurasi dengan subnet privat dalam minimal dua Availability Zone di Wilayah tersebut. Karena AWS CloudHSM tidak didukung di semua Availability Zone, sebaiknya Anda membuat subnet privat di semua Availability Zone di wilayah tersebut. Anda tidak dapat mengonfigurasi ulang subnet untuk klaster yang ada, tetapi Anda dapat membuat sebuah klaster dari cadangan dengan subnet yang berbeda dalam konfigurasi klaster.

  > **Important**
  >
  > Setelah Anda membuat toko AWS CloudHSM kunci Anda, jangan hapus salah satu subnet pribadi yang dikonfigurasi untuk AWS CloudHSM klasternya. Jika AWS KMS tidak dapat menemukan semua subnet dalam konfigurasi klaster, upaya untuk terhubung ke penyimpanan kunci kustom mengalami kegagalan dengan status kesalahan koneksi `SUBNET_NOT_FOUND`. Untuk detail selengkapnya, lihat Cara memperbaiki kegagalan koneksi.

- Grup keamanan untuk cluster (`cloudhsm-cluster-<cluster-id>-sg`) harus menyertakan aturan masuk dan aturan keluar yang memungkinkan lalu lintas TCP pada port 2223-2225. Sumber dalam aturan masuk dan Tujuan dalam aturan keluar harus sesuai dengan ID grup keamanan. Aturan ini ditetapkan secara default saat Anda membuat klaster. Jangan menghapus atau mengubah aturan tersebut.
- Klaster harus berisi setidaknya dua HSM aktif di Availability Zone yang berbeda. Untuk memverifikasi jumlah HSM, gunakan AWS CloudHSM konsol atau DescribeClustersoperasi. Jika perlu, Anda dapat menambahkan HSM.

Temukan sertifikat jangkar tepercaya

Saat membuat penyimpanan kunci kustom, Anda harus mengunggah sertifikat jangkar kepercayaan untuk AWS CloudHSM AWS KMS klaster. AWS KMSmembutuhkan sertifikat jangkar kepercayaan untuk menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klaster terkaitnya.

Setiap klaster AWS CloudHSM aktif memiliki sebuah sertifikat jangkar tepercaya. Ketika menginisialisasi klaster, Anda menghasilkan sertifikat ini, menyimpannya dalam file `customerCA.crt`, dan menyalinnya ke host yang terhubung ke klaster.

Buat pengguna kripto **`kmsuser`** untuk AWS KMS

Untuk mengelola toko AWS CloudHSM kunci Anda, masuk AWS KMS ke akun pengguna `kmsuser` kripto (CU) di klaster yang dipilih. Sebelum Anda membuat toko AWS CloudHSM kunci Anda, Anda harus membuat `kmsuser` CU. Kemudian ketika Anda membuat toko AWS CloudHSM kunci Anda, Anda memberikan kata sandi `kmsuser` untukAWS KMS. Setiap kali Anda menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM cluster terkaitAWS KMS, masuk sebagai `kmsuser` dan memutar kata sandi `kmsuser`

> **Important**
>
> Jangan menentukan opsi `2FA` saat Anda membuat CU `kmsuser`. Jika Anda melakukannya, AWS KMS tidak dapat masuk dan toko AWS CloudHSM kunci Anda tidak dapat terhubung ke AWS CloudHSM cluster ini. Setelah Anda menentukan 2FA, Anda tidak dapat membatalkannya. Sebaliknya, Anda harus menghapus CU dan membuatnya ulang.

Untuk membuat CU `kmsuser`, gunakan prosedur berikut.

1. Mulai cloudhsm_mgmt_util seperti yang dijelaskan dalam topik Memulai dengan CloudHSM Management Utility (CMU) dari Panduan Pengguna. AWS CloudHSM
2. Gunakan perintah createUser di cloudhsm_mgmt_util untuk membuat CU dengan nama `kmsuser`. Kata sandi harus berisi 7–32 karakter alfanumerik. Kata sandi peka huruf besar/kecil dan tidak dapat berisi karakter khusus.

   Misalnya, perintah contoh berikut membuat CU `kmsuser` dengan kata sandi `kmsPswd`.

```
aws-cloudhsm> createUser CU kmsuser kmsPswd
```

Buat toko AWS CloudHSM kunci (konsol)

Saat Anda membuat penyimpanan AWS CloudHSM kunci diAWS Management Console, Anda dapat menambahkan dan membuat prasyarat sebagai bagian dari alur kerja Anda. Namun, prosesnya lebih cepat saat Anda merakitnya terlebih dahulu.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.
4. Pilih Buat toko kunci.
5. Masukkan nama yang mudah diingat untuk penyimpanan kunci kustom. Nama harus unik di antara semua toko kunci khusus di akun Anda.

   > **Important**
   >
   > Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

6. Pilih AWS CloudHSM cluster untuk penyimpanan AWS CloudHSM kunci. Atau, untuk membuat klaster AWS CloudHSM baru, pilih tautan Buat klaster AWS CloudHSM.

   Menu menampilkan AWS CloudHSM cluster di akun dan wilayah Anda yang belum dikaitkan dengan toko AWS CloudHSM kunci. Klaster harus memenuhi persyaratan untuk asosiasi dengan penyimpanan kunci kustom.
7. Pilih Pilih file, lalu unggah sertifikat jangkar kepercayaan untuk AWS CloudHSM klaster yang Anda pilih. Ini adalah file `customerCA.crt` yang Anda buat saat Anda menginisialisasi klaster.
8. Masukkan kata sandi dari pengguna kripto (CU) `kmsuser` yang Anda buat dalam klaster yang dipilih.
9. Pilih Buat.

Ketika prosedur berhasil, toko AWS CloudHSM kunci baru muncul di daftar toko AWS CloudHSM utama di akun dan wilayah. Jika prosedur gagal, muncul pesan kesalahan yang menjelaskan

masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Memecahkan masalah penyimpanan kunci kustom.

Jika Anda mencoba membuat penyimpanan AWS CloudHSM kunci dengan semua nilai properti yang sama dengan penyimpanan AWS CloudHSM kunci terputus yang ada, AWS KMS tidak membuat penyimpanan AWS CloudHSM kunci baru, dan itu tidak memunculkan pengecualian atau menampilkan kesalahan. Sebaliknya, AWS KMS mengenali duplikat sebagai konsekuensi yang mungkin dari percobaan ulang, dan mengembalikan ID dari penyimpanan kunci yang adaAWS CloudHSM.

Berikutnya: Toko AWS CloudHSM kunci baru tidak terhubung secara otomatis. Sebelum Anda dapat membuat AWS KMS keys di toko AWS CloudHSM kunci, Anda harus menghubungkan toko kunci khusus ke AWS CloudHSM cluster terkait.

Buat toko AWS CloudHSM kunci (API)

Anda dapat menggunakan CreateCustomKeyStoreoperasi untuk membuat toko AWS CloudHSM kunci baru yang terkait dengan AWS CloudHSM cluster di akun dan Wilayah. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Operasi `CreateCustomKeyStore` memerlukan nilai parameter berikut.

- CustomKeyStoreName — Nama yang ramah untuk toko kunci khusus yang unik di akun.

  > **Important**
  >
  > Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

- CloudHsmClusterId — ID cluster AWS CloudHSM cluster yang memenuhi persyaratan untuk penyimpanan AWS CloudHSM kunci.
- KeyStorePassword — Kata sandi akun `kmsuser` CU di cluster yang ditentukan.
- TrustAnchorCertificate — Isi `customerCA.crt` file yang Anda buat saat Anda menginisialisasi cluster.

Contoh berikut menggunakan ID klaster fiktif. Sebelum menjalankan perintah, ganti dengan ID klaster yang valid.

```
$ 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>
```

Jika Anda menggunakan AWS CLI, Anda dapat menentukan file sertifikat jangkar tepercaya, bukan kontennya. Dalam contoh berikut, file `customerCA.crt` tidak berada dalam direktori akar.

```
$ 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
```

Saat operasi berhasil, `CreateCustomKeyStore` mengembalikan ID penyimpanan kunci kustom, seperti yang ditunjukkan dalam contoh tanggapan berikut.

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Jika operasi gagal, perbaiki kesalahan yang ditunjukkan oleh pengecualian, dan coba lagi. Untuk bantuan tambahan, lihat Memecahkan masalah penyimpanan kunci kustom.

Jika Anda mencoba membuat penyimpanan AWS CloudHSM kunci dengan semua nilai properti yang sama dengan penyimpanan AWS CloudHSM kunci terputus yang ada, AWS KMS tidak membuat penyimpanan AWS CloudHSM kunci baru, dan itu tidak memunculkan pengecualian atau menampilkan kesalahan. Sebaliknya, AWS KMS mengenali duplikat sebagai konsekuensi yang mungkin dari percobaan ulang, dan mengembalikan ID dari penyimpanan kunci yang adaAWS CloudHSM.

Berikutnya: Untuk menggunakan toko AWS CloudHSM kunci, hubungkan ke AWS CloudHSM klaster nya.

Melihat toko AWS CloudHSM kunci

Anda dapat melihat toko AWS CloudHSM utama di setiap akun dan Wilayah dengan menggunakan AWS KMS konsol atau DescribeCustomKeyStoresoperasi.

Lihat juga:

- Melihat toko kunci eksternal
- Melihat kunci KMS di toko AWS CloudHSM kunci
- Logging panggilan AWS KMS API dengan AWS CloudTrail

Topik

- Lihat toko AWS CloudHSM kunci (konsol)
- Lihat toko AWS CloudHSM kunci (API)

Lihat toko AWS CloudHSM kunci (konsol)

Ketika Anda melihat toko-toko AWS CloudHSM utama diAWS Management Console, Anda dapat melihat yang berikut:

- Nama dan ID toko kunci kustom
- ID klaster AWS CloudHSM yang terkait
- Jumlah HSM di klaster.
- Status koneksi saat ini

Nilai status koneksi (Status) dari Terputus menunjukkan bahwa penyimpanan kunci khusus baru dan belum pernah terhubung, atau sengaja terputus dari klasternya. AWS CloudHSM Namun, jika upaya Anda untuk menggunakan kunci KMS di penyimpanan kunci kustom yang terhubung gagal, itu mungkin menunjukkan masalah dengan penyimpanan kunci khusus atau AWS CloudHSM klasternya. Untuk bantuan, lihat Cara memperbaiki kunci KMS yang gagal.

Untuk melihat toko AWS CloudHSM utama di akun dan Wilayah tertentu, gunakan prosedur berikut.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.

Untuk menyesuaikan tampilan, klik ikon roda gigi yang muncul di bawah tombol Buat penyimpanan kunci.

Lihat toko AWS CloudHSM kunci (API)

Untuk melihat toko AWS CloudHSM utama Anda, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah. Namun Anda dapat menggunakan parameter `CustomKeyStoreId` atau `CustomKeyStoreName` (tetapi tidak keduanya) untuk membatasi output ke penyimpanan kunci kustom tertentu. Untuk penyimpanan AWS CloudHSM kunci, output terdiri dari ID dan nama penyimpanan kunci kustom, jenis penyimpanan kunci kustom, ID AWS CloudHSM cluster terkait, dan status koneksi. Jika status koneksi menunjukkan kesalahan, output juga menyertakan kode kesalahan yang menjelaskan alasan kesalahan.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Misalnya, perintah berikut mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah. Anda dapat menggunakan parameter `Limit` dan `Marker` ke halaman melalui penyimpanan kunci kustom dalam output.

```
$ aws kms describe-custom-key-stores
```

Contoh perintah berikut menggunakan parameter `CustomKeyStoreName` untuk mendapatkan hanya penyimpanan kunci kustom dengan nama `ExampleCloudHSMKeyStore` yang mudah diingat. Anda dapat menggunakan parameter `CustomKeyStoreName` atau `CustomKeyStoreId` (tetapi tidak keduanya) di setiap perintah.

Contoh output berikut merupakan penyimpanan AWS CloudHSM kunci yang terhubung ke AWS CloudHSM cluster nya.

> **Note**
>
> `CustomKeyStoreType`Bidang ditambahkan ke `DescribeCustomKeyStores` respons untuk membedakan toko AWS CloudHSM kunci dari toko kunci eksternal.

```
$ 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>"
        }
    ]
}
```

`ConnectionState` dari `Disconnected` menunjukkan bahwa penyimpanan kunci kustom belum pernah terhubung atau itu sengaja terputus dari klaster AWS CloudHSM-nya. Namun, jika upaya untuk menggunakan kunci KMS di penyimpanan AWS CloudHSM kunci yang terhubung gagal, itu mungkin menunjukkan masalah dengan penyimpanan AWS CloudHSM kunci atau AWS CloudHSM klasternya. Untuk bantuan, lihat Cara memperbaiki kunci KMS yang gagal.

Jika `ConnectionState` dari penyimpanan kunci kustom adalah `FAILED`, respons `DescribeCustomKeyStores` mencakup elemen `ConnectionErrorCode` yang menjelaskan alasan kesalahan.

Sebagai contoh, dalam output berikut, nilai `INVALID_CREDENTIALS` menunjukkan bahwa koneksi penyimpanan kunci kustom gagal karena kata sandi `kmsuser` tidak valid. Untuk bantuan dengan hal ini dan kegagalan kesalahan koneksi lainnya, lihat Memecahkan masalah penyimpanan kunci kustom.

```
$ 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>"
        }
    ]
}
```

Mengedit pengaturan toko AWS CloudHSM kunci

Anda dapat mengubah pengaturan toko AWS CloudHSM kunci yang ada. Toko kunci khusus harus terputus AWS CloudHSM klasternya.

Untuk mengedit pengaturan penyimpanan AWS CloudHSM kunci:

1. Putuskan koneksi penyimpanan kunci kustom dari klaster AWS CloudHSM-nya. Sementara toko kunci khusus terputus, Anda tidak dapat membuat AWS KMS keys(kunci KMS) di toko kunci khusus dan Anda tidak dapat menggunakan kunci KMS yang dikandungnya untuk operasi kriptografi.
2. Edit satu atau beberapa pengaturan penyimpanan AWS CloudHSM utama.
3. Sambungkan kembali penyimpanan kunci kustom ke klaster AWS CloudHSM-nya.

Anda dapat mengedit pengaturan berikut di penyimpanan kunci kustom:

Nama penyimpanan kunci kustom yang mudah diingat.

Masukkan nama baru yang mudah diingat. Nama baru harus unik di antara semua toko kunci khusus di AndaAkun AWS.

> **Important**
>
> Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

ID klaster dari klaster AWS CloudHSM terkait.

Edit nilai ini untuk menggantikan klaster AWS CloudHSM yang terkait untuk yang asli. Anda dapat menggunakan fitur ini untuk memperbaiki penyimpanan kunci kustom jika klaster AWS CloudHSM-nya rusak atau dihapus.

Tentukan klaster AWS CloudHSM yang berbagi riwayat pencadangan dengan klaster asli dan memenuhi persyaratan untuk asosiasi dengan penyimpanan kunci kustom, termasuk dua HSM aktif di Availability Zone yang berbeda. Klaster yang berbagi riwayat pencadangan memiliki sertifikat klaster yang sama. Untuk melihat sertifikat cluster cluster, gunakan DescribeClustersoperasi. Anda tidak dapat menggunakan fitur edit untuk mengaitkan penyimpanan kunci kustom dengan klaster AWS CloudHSM yang tidak terkait.

Kata sandi saat ini dari pengguna kripto (CU) **`kmsuser`**.

Beri tahu kata sandi AWS KMS saat ini dari CU `kmsuser` di klaster AWS CloudHSM. Tindakan ini tidak mengubah kata sandi CU `kmsuser` di klaster AWS CloudHSM.

Jika Anda mengubah kata sandi CU `kmsuser` di klaster AWS CloudHSM, gunakan fitur ini untuk memberi tahu AWS KMS kata sandi `kmsuser` baru. Jika tidak, AWS KMS tidak dapat login ke klaster dan semua upaya untuk menghubungkan penyimpanan kunci kustom untuk klaster mengalami kegagalan.

Topik

- Mengedit toko AWS CloudHSM kunci (konsol)
- Mengedit toko AWS CloudHSM kunci (API)

Mengedit toko AWS CloudHSM kunci (konsol)

Saat Anda mengedit penyimpanan AWS CloudHSM kunci, Anda dapat mengubah salah satu atau nilai yang dapat dikonfigurasi.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.
4. Pilih baris toko AWS CloudHSM kunci yang ingin Anda edit.

   Jika nilai di kolom status Koneksi tidak Terputus, Anda harus memutuskan penyimpanan kunci khusus sebelum Anda dapat mengeditnya. (Dari menu Key store actions, pilih Disconnect.)

   Sementara toko AWS CloudHSM kunci terputus, Anda dapat mengelola toko AWS CloudHSM kunci dan kunci KMS-nya, tetapi Anda tidak dapat membuat atau menggunakan kunci KMS di toko kunci. AWS CloudHSM
5. Dari menu Key store actions, pilih Edit.
6. Lakukan satu atau beberapa tindakan berikut.

   - Ketikkan nama yang mudah diingat untuk penyimpanan kunci kustom.
   - Ketikkan ID klaster dari klaster AWS CloudHSM yang terkait.

- Ketikkan kata sandi saat ini dari pengguna kripto `kmsuser` di klaster AWS CloudHSM yang terkait.

7. Pilih Simpan.

   Ketika prosedur berhasil, suatu pesan akan menjelaskan pengaturan yang Anda diedit. Ketika prosedur gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Memecahkan masalah penyimpanan kunci kustom.

8. Hubungkan kembali penyimpanan kunci kustom.

   Untuk menggunakan toko AWS CloudHSM kunci, Anda harus menghubungkannya kembali setelah mengedit. Anda dapat membiarkan toko AWS CloudHSM kunci terputus. Tetapi saat terputus, Anda tidak dapat membuat kunci KMS di toko AWS CloudHSM kunci atau menggunakan kunci KMS di toko kunci dalam operasi AWS CloudHSM kriptografi.

Mengedit toko AWS CloudHSM kunci (API)

Untuk mengubah properti toko AWS CloudHSM kunci, gunakan UpdateCustomKeyStoreoperasi. Anda dapat mengubah beberapa properti dari penyimpanan kunci kustom dalam perintah yang sama. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti. Untuk memverifikasi bahwa perubahannya efektif, gunakan DescribeCustomKeyStoresoperasi.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Mulailah dengan menggunakan DisconnectCustomKeyStoreuntuk memutuskan penyimpanan kunci kustom dari AWS CloudHSM klasternya. Ganti contoh ID penyimpanan kunci kustom, cks-1234567890abcdef0, dengan ID yang sebenarnya.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Contoh pertama digunakan UpdateCustomKeyStoreuntuk mengubah nama ramah dari toko AWS CloudHSM kunci menjadi`DevelopmentKeys`. Perintah menggunakan `CustomKeyStoreId` parameter untuk mengidentifikasi penyimpanan AWS CloudHSM kunci dan `CustomKeyStoreName` untuk menentukan nama baru untuk toko kunci kustom.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name DevelopmentKeys
```

Contoh berikut mengubah cluster yang terkait dengan penyimpanan AWS CloudHSM kunci ke cadangan lain dari cluster yang sama. Perintah menggunakan `CustomKeyStoreId` parameter untuk mengidentifikasi penyimpanan AWS CloudHSM kunci dan `CloudHsmClusterId` parameter untuk menentukan ID cluster baru.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Contoh berikut memberi tahu AWS KMS bahwa kata sandi `kmsuser` saat ini adalah `ExamplePassword`. Perintah menggunakan `CustomKeyStoreId` parameter untuk mengidentifikasi penyimpanan AWS CloudHSM kunci dan `KeyStorePassword` parameter untuk menentukan kata sandi saat ini.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

Perintah terakhir menghubungkan kembali penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klasternya. Anda dapat meninggalkan penyimpanan kunci khusus dalam keadaan terputus, tetapi Anda harus menghubungkannya sebelum Anda dapat membuat kunci KMS baru atau menggunakan kunci KMS yang ada untuk operasi kriptografi. Ganti contoh ID penyimpanan kunci kustom dengan ID yang sebenarnya.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Menghubungkan dan memutuskan sambungan toko AWS CloudHSM kunci

Toko AWS CloudHSM kunci baru tidak terhubung. Sebelum Anda dapat membuat dan menggunakan AWS KMS keys di toko AWS CloudHSM kunci Anda, Anda harus menghubungkannya ke AWS CloudHSM cluster terkait. Anda dapat menghubungkan dan memutuskan penyimpanan AWS CloudHSM kunci Anda kapan saja, dan melihat status koneksinya.

Anda tidak diharuskan untuk menghubungkan toko AWS CloudHSM kunci Anda. Anda dapat meninggalkan toko AWS CloudHSM kunci dalam keadaan terputus tanpa batas waktu dan menghubungkannya hanya ketika Anda perlu menggunakannya. Namun, Anda mungkin ingin menguji koneksi secara berkala untuk memverifikasi bahwa pengaturan sudah benar dan dapat tersambung.

> **Note**
>
> AWS CloudHSMtoko kunci memiliki status `DISCONNECTED` koneksi hanya ketika toko kunci tidak pernah terhubung atau Anda secara eksplisit memutuskannya. Jika status koneksi penyimpanan AWS CloudHSM kunci Anda `CONNECTED` tetapi Anda mengalami masalah dalam menggunakannya, pastikan AWS CloudHSM klaster terkaitnya aktif dan berisi setidaknya satu HSM aktif. Untuk bantuan dengan kegagalan koneksi, lihat the section called "Memecahkan masalah penyimpanan kunci kustom".

Topik

Menghubungkan toko AWS CloudHSM kunci

Saat Anda menghubungkan penyimpanan AWS CloudHSM kunci, AWS KMS temukan AWS CloudHSM klaster terkait, sambungkan ke sana, masuk ke AWS CloudHSM klien sebagai pengguna `kmsuser` kripto (CU), dan kemudian memutar `kmsuser` kata sandi. AWS KMStetap masuk ke AWS CloudHSM klien selama toko AWS CloudHSM kunci terhubung.

Untuk membuat koneksi, AWS KMS membuat grup keamanan bernama `kms-<custom key store ID>` di virtual private cloud (VPC) klaster. Grup keamanan memiliki aturan tunggal yang memungkinkan lalu lintas masuk dari grup keamanan cluster. AWS KMS juga membuat antarmuka jaringan elastis (ENI) di setiap Availability Zone dari subnet privat untuk klaster. AWS KMS menambahkan ENI ke grup keamanan `kms-<cluster ID>` dan grup keamanan untuk klaster. Deskripsi dari masing-masing ENI adalah `KMS managed ENI for cluster <cluster-ID>`.

Proses koneksi dapat memakan waktu yang lama untuk selesai; hingga 20 menit.

Sebelum Anda menghubungkan toko AWS CloudHSM kunci, verifikasi bahwa itu memenuhi persyaratan.

- Klaster AWS CloudHSM yang terkait harus memuat minimal satu HSM aktif. Untuk menemukan jumlah HSM di cluster, lihat cluster di AWS CloudHSM konsol atau gunakan DescribeClustersoperasi. Jika perlu, Anda dapat menambahkan HSM.
- Cluster harus memiliki akun pengguna `kmsuser` kripto (CU), tetapi CU tersebut tidak dapat masuk ke cluster saat Anda menghubungkan penyimpanan AWS CloudHSM kunci. Untuk bantuan dengan logout, lihat Cara logout dan menghubungkan kembali.
- Status koneksi dari toko AWS CloudHSM kunci tidak dapat `DISCONNECTING` atau`FAILED`. Untuk melihat status koneksi, gunakan AWS KMS konsol atau DescribeCustomKeyStoresrespons. Jika status koneksi`FAILED`, lepaskan penyimpanan kunci khusus, perbaiki masalahnya, lalu sambungkan.

Untuk bantuan dengan kegagalan koneksi, lihat Cara memperbaiki kegagalan koneksi.

Ketika toko AWS CloudHSM kunci Anda terhubung, Anda dapat membuat kunci KMS di dalamnya dan menggunakan kunci KMS yang ada dalam operasi kriptografi.

Memutuskan sambungan toko AWS CloudHSM kunci

Saat Anda memutuskan sambungan penyimpanan AWS CloudHSM kunci, AWS KMS keluar dari AWS CloudHSM klien, terputus dari AWS CloudHSM cluster terkait, dan menghapus infrastruktur jaringan yang dibuatnya untuk mendukung koneksi.

Sementara toko AWS CloudHSM kunci terputus, Anda dapat mengelola toko AWS CloudHSM kunci dan kunci KMS-nya, tetapi Anda tidak dapat membuat atau menggunakan kunci KMS di toko kunci. AWS CloudHSM Status koneksi dari toko kunci adalah `DISCONNECTED` dan status kunci kunci KMS di toko kunci kustom adalah`Unavailable`, kecuali mereka`PendingDeletion`. Anda dapat menghubungkan kembali toko AWS CloudHSM kunci kapan saja.

Saat Anda memutuskan penyimpanan kunci khusus, kunci KMS di toko kunci menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhiLayanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Untuk detailnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

> **Note**
>
> Sementara toko kunci khusus terputus, semua upaya untuk membuat kunci KMS di toko kunci khusus atau menggunakan kunci KMS yang ada dalam operasi kriptografi akan gagal. Tindakan ini dapat mencegah pengguna menyimpan dan mengakses data sensitif.

Untuk memperkirakan efek pemutusan toko kunci kustom Anda dengan lebih baik, identifikasi kunci KMS di toko kunci khusus dan tentukan penggunaannya di masa lalu.

Anda dapat memutuskan sambungan penyimpanan AWS CloudHSM kunci karena alasan seperti berikut:

- Untuk memutar kata sandi **`kmsuser`**. AWS KMS mengubah kata sandi `kmsuser` setiap kali terhubung ke klaster AWS CloudHSM. Untuk memberlakukan rotasi kata sandi, cukup putuskan koneksi dan hubungkan kembali.
- Untuk mengaudit materi kunci untuk kunci KMS di AWS CloudHSM cluster. Ketika Anda memutuskan koneksi penyimpanan kunci kustom, AWS KMS logout dari pengguna kripto `kmsuser`dalam klien AWS CloudHSM. Ini memungkinkan Anda untuk masuk ke cluster sebagai `kmsuser` CU dan mengaudit dan mengelola materi utama untuk kunci KMS.
- Untuk segera menonaktifkan semua kunci KMS di toko AWS CloudHSM kunci. Anda dapat menonaktifkan dan mengaktifkan kembali kunci KMS di toko AWS CloudHSM kunci dengan menggunakan AWS Management Console atau operasi. DisableKey Operasi ini selesai dengan cepat, tetapi mereka bertindak pada satu kunci KMS pada satu waktu. Memutuskan sambungan penyimpanan AWS CloudHSM kunci segera mengubah status kunci dari semua kunci KMS di toko AWS CloudHSM kunci`Unavailable`, yang mencegahnya digunakan dalam operasi kriptografi apa pun.
- Untuk memperbaiki upaya koneksi yang gagal. Jika upaya untuk menghubungkan penyimpanan AWS CloudHSM kunci gagal (status koneksi penyimpanan kunci kustom`FAILED`), Anda harus memutuskan sambungan penyimpanan AWS CloudHSM kunci sebelum Anda mencoba menghubungkannya lagi.

Hubungkan toko AWS CloudHSM kunci (konsol)

Untuk menghubungkan toko AWS CloudHSM kunci diAWS Management Console, mulailah dengan memilih toko AWS CloudHSM kunci dari halaman Toko kunci kustom. Proses koneksi bisa memakan waktu hingga 20 menit untuk menyelesaikannya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.
4. Pilih baris toko AWS CloudHSM kunci yang ingin Anda sambungkan.

   Jika status koneksi penyimpanan AWS CloudHSM kunci Gagal, Anda harus memutuskan penyimpanan kunci khusus sebelum Anda menghubungkannya.
5. Dari menu Key Store Actions, pilih Connect.

AWS KMS memulai proses menghubungkan penyimpanan kunci kustom Anda. Itu menemukan klaster AWS CloudHSM terkait, membangun infrastruktur jaringan yang diperlukan, menghubungkan ke sana, login ke klaster AWS CloudHSM sebagai CU `kmsuser`, dan memutar kata sandi `kmsuser`. Ketika operasi selesai, status koneksi berubah menjadi Terhubung.

Jika operasi gagal, muncul pesan kesalahan yang menjelaskan alasan kegagalan. Sebelum Anda mencoba menghubungkan lagi, lihat status koneksi toko AWS CloudHSM kunci Anda. Jika Gagal, Anda harus memutuskan penyimpanan kunci khusus sebelum Anda menghubungkannya lagi. Jika Anda memerlukan bantuan, lihat Memecahkan masalah penyimpanan kunci kustom.

Berikutnya:the section called “Membuat kunci KMS di toko AWS CloudHSM kunci”.

Menghubungkan penyimpanan kunci kustom (API)

Untuk menghubungkan toko AWS CloudHSM kunci yang terputus, gunakan ConnectCustomKeyStoreoperasi. AWS CloudHSMCluster terkait harus berisi setidaknya satu HSM aktif dan status koneksi tidak dapat`FAILED`.

Proses koneksi memerlukan waktu yang lama untuk selesai; hingga 20 menit. Kecuali mengalami kegagalan dengan cepat, operasi tersebut mengembalikan respons HTTP 200 dan objek JSON tanpa properti. Namun, respons awal ini tidak menunjukkan bahwa koneksi berhasil. Untuk menentukan status koneksi penyimpanan kunci kustom, lihat DescribeCustomKeyStoresresponsnya.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Untuk mengidentifikasi toko AWS CloudHSM kunci, gunakan ID toko kunci kustom. Anda dapat menemukan ID di halaman toko kunci kustom di konsol atau dengan menggunakan

DescribeCustomKeyStoresoperasi tanpa parameter. Sebelum menjalankan contoh ini, ganti contoh ID dengan yang valid.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Untuk memverifikasi bahwa toko AWS CloudHSM kunci terhubung, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah Anda. Namun Anda dapat menggunakan parameter `CustomKeyStoreId` atau `CustomKeyStoreName` (tetapi tidak keduanya) untuk membatasi respons ke penyimpanan kunci kustom tertentu. Nilai `ConnectionState` dari `CONNECTED` menunjukkan bahwa penyimpanan kunci kustom terhubung ke klaster AWS CloudHSM.

> **Note**
>
> `CustomKeyStoreType`Bidang ditambahkan ke `DescribeCustomKeyStores` respons untuk membedakan toko AWS CloudHSM kunci dari toko kunci eksternal.

```
$ 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"
   ],
}
```

Jika nilai `ConnectionState` gagal, elemen `ConnectionErrorCode` menunjukkan alasan kegagalan. Dalam kasus ini, AWS KMS tidak dapat menemukan klaster AWS CloudHSM di akun Anda dengan ID klaster `cluster-1a23b4cdefg`. Jika Anda menghapus klaster, Anda dapat memulihkannya dari cadangan dari klaster asli dan mengedit ID klaster untuk penyimpanan kunci kustom. Untuk bantuan menanggapi kode kesalahan koneksi, lihatCara memperbaiki kegagalan koneksi.

```
$ 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"
   ],
}
```

Berikutnya:Membuat kunci KMS di toko AWS CloudHSM kunci.

Putuskan sambungan toko AWS CloudHSM kunci (konsol)

Untuk memutuskan penyimpanan AWS CloudHSM kunci yang terhubung diAWS Management Console, mulailah dengan memilih toko AWS CloudHSM kunci dari halaman Toko Kunci Kustom.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.
4. Pilih baris toko kunci eksternal yang ingin Anda putuskan.
5. Dari menu Key Store Actions, pilih Disconnect.

Ketika operasi selesai, status koneksi berubah dari Memutuskan sambungan ke Terputus. Jika operasi gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Memecahkan masalah penyimpanan kunci kustom.

Putuskan sambungan toko AWS CloudHSM kunci (API)

Untuk memutuskan sambungan AWS CloudHSM kunci yang terhubung, gunakan DisconnectCustomKeyStoreoperasi. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Contoh ini memutus penyimpanan AWS CloudHSM kunci. Sebelum menjalankan contoh ini, ganti contoh ID dengan yang valid.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Untuk memverifikasi bahwa penyimpanan AWS CloudHSM kunci terputus, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah Anda. Namun Anda dapat menggunakan parameter `CustomKeyStoreId` dan `CustomKeyStoreName` (tetapi tidak keduanya) untuk membatasi respons ke penyimpanan kunci kustom tertentu. `ConnectionState`Nilai `DISCONNECTED` menunjukkan bahwa penyimpanan AWS CloudHSM kunci contoh ini tidak terhubung ke AWS CloudHSM klasternya.

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

Menghapus toko AWS CloudHSM kunci

Saat Anda menghapus penyimpanan AWS CloudHSM kunci, AWS KMS menghapus semua metadata tentang penyimpanan AWS CloudHSM kunci dari KMS, termasuk informasi tentang hubungannya dengan klaster. AWS CloudHSM Operasi ini tidak memengaruhi klaster AWS CloudHSM, HSM, atau penggunanya. Anda dapat membuat penyimpanan AWS CloudHSM kunci baru yang terkait dengan AWS CloudHSM cluster yang sama, tetapi Anda tidak dapat membatalkan operasi penghapusan.

Anda hanya dapat menghapus penyimpanan AWS CloudHSM kunci yang terputus dari AWS CloudHSM klasternya dan tidak berisi apa punAWS KMS keys. Sebelum Anda menghapus penyimpanan kunci kustom, lakukan hal berikut:

- Verifikasi bahwa Anda tidak perlu menggunakan salah satu kunci KMS di toko kunci untuk operasi kriptografi apa pun. Kemudian jadwalkan penghapusan semua kunci KMS dari toko kunci. Untuk bantuan menemukan kunci KMS di toko AWS CloudHSM kunci, lihatTemukan kunci KMS di toko AWS CloudHSM kunci.
- Konfirmasikan bahwa semua kunci KMS telah dihapus. Untuk melihat kunci KMS di toko AWS CloudHSM kunci, lihatMelihat kunci KMS di toko AWS CloudHSM kunci.
- Putuskan sambungan penyimpanan AWS CloudHSM kunci dari AWS CloudHSM klasternya.

Alih-alih menghapus penyimpanan AWS CloudHSM kunci, pertimbangkan untuk memutuskannya dari cluster terkaitAWS CloudHSM. Sementara toko AWS CloudHSM kunci terputus, Anda dapat mengelola toko AWS CloudHSM kunci dan nyaAWS KMS keys. Tetapi Anda tidak dapat membuat atau menggunakan kunci KMS di toko AWS CloudHSM kunci. Anda dapat menghubungkan kembali toko AWS CloudHSM kunci kapan saja.

Topik

- Hapus toko AWS CloudHSM kunci (konsol)
- Hapus toko AWS CloudHSM kunci (API)

Hapus toko AWS CloudHSM kunci (konsol)

Untuk menghapus toko AWS CloudHSM kunci diAWS Management Console, mulailah dengan memilih toko AWS CloudHSM kunci dari halaman Toko kunci kustom.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, toko AWS CloudHSM utama.
4. Temukan baris yang mewakili penyimpanan AWS CloudHSM kunci yang ingin Anda hapus. Jika status koneksi penyimpanan AWS CloudHSM kunci tidak Terputus, Anda harus memutuskan penyimpanan AWS CloudHSM kunci sebelum Anda menghapusnya.
5. Dari menu Key Store Actions, pilih Delete.

Ketika operasi selesai, pesan sukses muncul dan toko AWS CloudHSM kunci tidak lagi muncul di daftar toko utama. Jika operasi gagal, muncul pesan kesalahan yang menjelaskan masalah dan

memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Memecahkan masalah penyimpanan kunci kustom.

Hapus toko AWS CloudHSM kunci (API)

Untuk menghapus toko AWS CloudHSM kunci, gunakan DeleteCustomKeyStoreoperasi. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti.

Untuk memulai, verifikasi bahwa toko AWS CloudHSM kunci tidak mengandung apa punAWS KMS keys. Anda tidak dapat menghapus toko kunci khusus yang berisi kunci KMS. Perintah contoh pertama menggunakan ListKeysdan DescribeKeymencari AWS KMS keys di toko AWS CloudHSM kunci dengan contoh *cks-1234567890abcdef0* ID toko kunci kustom. Dalam hal ini, perintah tidak mengembalikan kunci KMS apa pun. Jika ya, gunakan ScheduleKeyDeletionoperasi untuk menjadwalkan penghapusan masing-masing tombol KMS.

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'
```

Selanjutnya, lepaskan AWS CloudHSM kunci toko. Perintah contoh ini menggunakan DisconnectCustomKeyStoreoperasi untuk memutuskan penyimpanan AWS CloudHSM kunci dari AWS CloudHSM klaster nya. Sebelum menjalankan perintah ini, ganti contoh ID penyimpanan kunci kustom dengan yang valid.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Setelah toko kunci khusus terputus, Anda dapat menggunakan DeleteCustomKeyStoreoperasi untuk menghapusnya.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Mengelola kunci KMS di toko kunci CloudHSM

Anda dapat membuat, melihat, mengelola, menggunakan, dan menjadwalkan penghapusan AWS KMS keys di toko AWS CloudHSM utama. Prosedur yang Anda gunakan sangat mirip dengan yang Anda gunakan untuk kunci KMS lainnya. Satu-satunya perbedaan adalah Anda menentukan penyimpanan AWS CloudHSM kunci saat Anda membuat kunci KMS. Kemudian, AWS KMS buat materi kunci yang tidak dapat diekstraksi untuk kunci KMS di AWS CloudHSM cluster yang terkait dengan penyimpanan kunci. AWS CloudHSM Ketika Anda menggunakan kunci KMS di toko AWS CloudHSM kunci, operasi kriptografi dilakukan di HSM di cluster.

Fitur yang didukung

Selain prosedur yang dibahas di bagian ini, Anda dapat melakukan hal berikut dengan kunci KMS di toko AWS CloudHSM kunci:

- Gunakan kebijakan utama, kebijakan IAM, dan hibah untuk mengotorisasi akses ke kunci KMS.
- Aktifkan dan nonaktifkan tombol KMS.
- Tetapkan tag dan buat alias, dan gunakan kontrol akses berbasis atribut (ABAC) untuk mengotorisasi akses ke kunci KMS.
- Gunakan kunci KMS untuk operasi kriptografi, termasuk mengenkripsi, mendekripsi, mengenkripsi ulang, dan menghasilkan kunci data.
- Gunakan kunci KMS dengan AWSlayanan yang terintegrasi dengan AWS KMS dan mendukung kunci yang dikelola pelanggan.
- Lacak penggunaan kunci KMS Anda di AWS CloudTraillog dan alat CloudWatch pemantauan Amazon.

Fitur yang tidak didukung

- AWS CloudHSMtoko kunci hanya mendukung kunci KMS enkripsi simetris. Anda tidak dapat membuat kunci HMAC KMS, kunci KMS asimetris, atau pasangan kunci data asimetris di penyimpanan kunci. AWS CloudHSM
- Anda tidak dapat mengimpor materi kunci ke kunci KMS di toko AWS CloudHSM kunci. AWS KMSmenghasilkan bahan kunci untuk kunci KMS di AWS CloudHSM cluster.
- Anda tidak dapat mengaktifkan atau menonaktifkan rotasi otomatis bahan kunci untuk kunci KMS di toko AWS CloudHSM kunci.

Topik

- Membuat kunci KMS di toko AWS CloudHSM kunci
- Melihat kunci KMS di toko AWS CloudHSM kunci
- Menggunakan kunci KMS di toko AWS CloudHSM kunci
- Menemukan kunci KMS dan bahan utama
- Penjadwalan penghapusan kunci KMS dari toko kunci AWS CloudHSM

Membuat kunci KMS di toko AWS CloudHSM kunci

Setelah Anda membuat toko AWS CloudHSM kunci, Anda dapat membuat AWS KMS keysdi toko kunci Anda. Mereka harus enkripsi simetris kunci KMS dengan bahan kunci yang AWS KMS menghasilkan. Anda tidak dapat membuat kunci KMS asimetris, kunci KMS HMAC atau kunci KMS dengan bahan kunci yang diimpor di toko kunci khusus. Selain itu, Anda tidak dapat menggunakan kunci KMS enkripsi simetris di toko kunci khusus untuk menghasilkan pasangan kunci data asimetris.

Untuk membuat kunci KMS di penyimpanan AWS CloudHSM kunci, penyimpanan AWS CloudHSM kunci harus terhubung ke AWS CloudHSM cluster terkait dan cluster harus berisi setidaknya dua HSM aktif di Availability Zone yang berbeda. Untuk menemukan status koneksi dan jumlah HSM, lihat halaman toko AWS CloudHSM utama di file. AWS Management Console Saat menggunakan operasi API, gunakan DescribeCustomKeyStoresoperasi untuk memverifikasi bahwa penyimpanan AWS CloudHSM kunci terhubung. Untuk memverifikasi jumlah HSM aktif di cluster dan Availability Zones mereka, gunakan AWS CloudHSM DescribeClustersoperasi.

Saat Anda membuat kunci KMS di toko AWS CloudHSM kunci Anda, AWS KMS buat kunci KMS. AWS KMS Tapi, itu menciptakan materi kunci untuk kunci KMS di AWS CloudHSM cluster terkait. Secara khusus, AWS KMS masuk ke dalam klaster sebagai CU `kmsuser` yang Anda buat. Kemudian

ia menciptakan kunci simetris Advanced Encryption Standard (AES) 256-bit yang persisten, tidak dapat diekstraksi, di cluster. AWS KMSmenetapkan nilai atribut label kunci, yang hanya terlihat di cluster, ke Amazon Resource Name (ARN) dari kunci KMS.

Ketika perintah berhasil, status kunci dari kunci KMS baru adalah `Enabled` dan asalnya adalah. `AWS_CLOUDHSM` Anda tidak dapat mengubah asal kunci KMS apa pun setelah Anda membuatnya. Saat Anda melihat kunci KMS di penyimpanan AWS CloudHSM kunci di AWS KMS konsol atau dengan menggunakan DescribeKeyoperasi, Anda dapat melihat properti tipikal, seperti ID kunci, status kunci, dan tanggal pembuatannya. Tetapi Anda juga dapat melihat ID penyimpanan kunci kustom dan (secara opsional) ID klaster AWS CloudHSM. Untuk rincian selengkapnya, lihat Melihat kunci KMS di toko AWS CloudHSM kunci.

Jika upaya Anda untuk membuat kunci KMS di toko AWS CloudHSM kunci Anda gagal, gunakan pesan kesalahan untuk membantu Anda menentukan penyebabnya. Ini mungkin menunjukkan bahwa penyimpanan AWS CloudHSM kunci tidak terhubung (`CustomKeyStoreInvalidStateException`) atau AWS CloudHSM cluster terkait tidak memiliki dua HSM aktif yang diperlukan untuk operasi ini (`CloudHsmClusterInvalidConfigurationException`). Untuk bantuan, lihat Memecahkan masalah penyimpanan kunci kustom.

Untuk contoh AWS CloudTrail log operasi yang membuat kunci KMS di toko AWS CloudHSM kunci, lihatCreateKey.

Topik

- Buat kunci KMS di toko AWS CloudHSM kunci (konsol)
- Buat kunci KMS di toko AWS CloudHSM kunci (API)

Buat kunci KMS di toko AWS CloudHSM kunci (konsol)

Gunakan prosedur berikut untuk membuat kunci KMS enkripsi simetris di toko AWS CloudHSM kunci.

> **Note**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.

2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Pilih Simetris.
6. Dalam Penggunaan kunci, opsi Enkripsi dan dekripsi dipilih untuk Anda. Jangan mengubahnya.
7. Pilih Opsi lanjutan.
8. Untuk asal bahan utama, pilih toko AWS CloudHSM kunci.

   Anda tidak dapat membuat kunci Multi-wilayah di toko AWS CloudHSM kunci.
9. Pilih Berikutnya.
10. Pilih toko AWS CloudHSM kunci untuk kunci KMS baru Anda. Untuk membuat toko AWS CloudHSM kunci baru, pilih Buat toko kunci khusus.

    Toko AWS CloudHSM kunci yang Anda pilih harus memiliki status Terhubung. Klaster AWS CloudHSM terkaitnya harus aktif dan berisi setidaknya dua HSM aktif di Availability Zone yang berbeda.

    Untuk bantuan menghubungkan toko AWS CloudHSM kunci, lihatMenghubungkan dan memutuskan sambungan toko AWS CloudHSM kunci. Untuk bantuan dengan menambahkan HSM, lihat Menambahkan HSM di Panduan Pengguna AWS CloudHSM.
11. Pilih Selanjutnya.
12. Ketik alias dan deskripsi opsional untuk kunci KMS.
13. (Opsional). Pada halaman Tambah Tag, tambahkan tag yang mengidentifikasi atau mengkategorikan kunci KMS Anda.

    Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS
14. Pilih Berikutnya.
15. Di bagian Administrator Kunci, pilih pengguna IAM dan peran yang dapat mengelola kunci KMS. Untuk informasi selengkapnya, lihat Mengizinkan administrator kunci mengelola kunci KMS.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

16. (Opsional) Untuk mencegah administrator kunci ini menghapus kunci KMS ini, kosongkan kotak di bagian bawah halaman untuk Izinkan administrator kunci untuk menghapus kunci ini.
17. Pilih Berikutnya.
18. Di bagian Akun ini, pilih pengguna IAM dan peran dalam hal ini Akun AWS yang dapat menggunakan kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Mengizinkan pengguna kunci menggunakan kunci KMS.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensi jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

19. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, di Akun AWS bagian Lain di bagian bawah halaman, pilih Tambahkan yang lain Akun AWS dan masukkan Akun AWS ID akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Administrator dari pihak lain juga Akun AWS harus mengizinkan akses ke kunci KMS dengan membuat kebijakan IAM untuk pengguna mereka. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

20. Pilih Selanjutnya.
21. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
22. Setelah selesai, pilih Selesai untuk membuat kunci.

Ketika prosedur berhasil, tampilan menunjukkan kunci KMS baru di toko AWS CloudHSM kunci yang Anda pilih. Saat Anda memilih nama atau alias kunci KMS baru, tab konfigurasi Kriptografi pada halaman detailnya menampilkan asal kunci KMS (AWS CloudHSM), nama, ID, dan jenis penyimpanan kunci khusus, dan ID cluster. AWS CloudHSM Jika prosedur gagal, muncul pesan kesalahan yang menjelaskan kegagalan.

> **Tip**
>
> Untuk mempermudah mengidentifikasi kunci KMS di toko kunci kustom, pada halaman Kunci yang dikelola Pelanggan, tambahkan kolom ID penyimpanan kunci kustom ke tampilan. Klik ikon roda gigi di kanan atas dan pilih ID penyimpanan kunci kustom. Untuk detailnya, lihat Menyesuaikan tabel kunci KMS Anda.

Buat kunci KMS di toko AWS CloudHSM kunci (API)

Untuk membuat AWS KMS key(kunci KMS) baru di toko AWS CloudHSM kunci Anda, gunakan CreateKeyoperasi. Gunakan parameter `CustomKeyStoreId` untuk mengidentifikasi penyimpanan kunci kustom Anda dan menentukan nilai `Origin` dari `AWS_CLOUDHSM`.

Anda mungkin juga ingin menggunakan parameter `Policy` untuk menentukan kebijakan kunci. Anda dapat mengubah kebijakan kunci (PutKeyPolicy) dan menambahkan elemen opsional, seperti deskripsi dan tag kapan saja.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Contoh berikut dimulai dengan panggilan ke DescribeCustomKeyStoresoperasi untuk memverifikasi bahwa penyimpanan AWS CloudHSM kunci terhubung ke AWS CloudHSM cluster terkait. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah Anda. Untuk menggambarkan hanya penyimpanan AWS CloudHSM kunci tertentu, gunakan `CustomKeyStoreId` atau `CustomKeyStoreName` parameternya (tetapi tidak keduanya).

Sebelum menjalankan perintah ini, ganti contoh ID penyimpanan kunci kustom dengan ID yang valid.

> **Note**
>
> Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

```
$ 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"
   ],
}
```

Contoh perintah berikutnya menggunakan DescribeClustersoperasi untuk memverifikasi bahwa AWS CloudHSM cluster yang terkait dengan `ExampleKeyStore` (cluster-1a23b4cdefg) memiliki setidaknya dua HSM aktif. Jika klaster memiliki kurang dari dua HSM, operasi `CreateKey` gagal.

```
$ 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"
        }
    ]
}
```

Perintah contoh ini menggunakan CreateKeyoperasi untuk membuat kunci KMS di toko AWS CloudHSM kunci. Untuk membuat kunci KMS di toko AWS CloudHSM kunci, Anda harus memberikan ID penyimpanan kunci kustom dari toko AWS CloudHSM kunci dan menentukan `Origin` nilai`AWS_CLOUDHSM`.

Tanggapan meliputi ID dari penyimpanan kunci kustom dan klaster AWS CloudHSM.

Sebelum menjalankan perintah ini, ganti contoh ID penyimpanan kunci kustom dengan ID yang valid.

```
$ 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"
    ]
  }
}
```

Melihat kunci KMS di toko AWS CloudHSM kunci

Untuk melihat AWS KMS keys di toko AWS CloudHSM kunci, gunakan teknik yang sama yang akan Anda gunakan untuk melihat kunci yang dikelola AWS KMS pelanggan. Untuk mempelajari dasar-dasarnya, lihat Melihat kunci. Untuk mengidentifikasi kunci di AWS CloudHSM klaster Anda yang berfungsi sebagai bahan utama untuk kunci KMS Anda, lihatMenemukan kunci KMS dan bahan utama. Untuk informasi tentang melihat AWS CloudTrail log yang merekam semua operasi API di penyimpanan kunci kustom, lihatLogging panggilan AWS KMS API dengan AWS CloudTrail.

Di AWS KMS konsol, kunci KMS di toko kunci kustom Anda ditampilkan di halaman kunci yang dikelola Pelanggan, bersama dengan semua kunci yang dikelola pelanggan lainnya di wilayah Anda Akun AWS dan wilayah.

Namun, nilai-nilai berikut khusus untuk kunci KMS di toko AWS CloudHSM kunci.

- Nama dan ID toko AWS CloudHSM kunci yang menyimpan kunci KMS.
- ID klaster dari klaster AWS CloudHSM terkait yang berisi material kunci mereka.
- `Origin`Nilai `AWS CloudHSM` di AWS KMS konsol atau `AWS_CLOUDHSM` respons API.
- Nilai status kunci dapat berupa `Unavailable`. Untuk bantuan menyelesaikan status, lihat Cara memperbaiki kunci KMS yang tidak tersedia.

Untuk melihat kunci KMS di toko AWS CloudHSM kunci (Konsol)

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilihan Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Di sudut kanan atas, pilih ikon roda gigi, pilih ikon roda gigi, pilih ID penyimpanan kunci kustom dan Asal, lalu pilih Konfirmasi.
5. Untuk mengidentifikasi kunci KMS di toko AWS CloudHSM kunci apa pun, cari kunci KMS dengan nilai Asal. AWS CloudHSM Untuk mengidentifikasi kunci KMS di toko AWS CloudHSM kunci tertentu, lihat nilai di kolom ID penyimpanan kunci kustom.
6. Pilih alias atau ID kunci kunci KMS di toko AWS CloudHSM kunci.

   Halaman ini menampilkan informasi terperinci tentang kunci KMS, termasuk Nama Sumber Daya Amazon (ARN), kebijakan kunci, dan tag.
7. Pilih tab Konfigurasi kriptografi. Tab ada di bawah bagian Konfigurasi umum.

   Bagian ini mencakup informasi tentang penyimpanan AWS CloudHSM kunci dan AWS CloudHSM cluster yang terkait dengan kunci KMS.

Untuk melihat kunci KMS di toko kunci kustom (API)

Anda menggunakan operasi AWS KMS API yang sama untuk melihat kunci KMS di penyimpanan AWS CloudHSM kunci yang akan Anda gunakan untuk kunci KMS apa pun, termasuk, ListKeysDescribeKey, dan. GetKeyPolicy Misalnya, `describe-key` operasi berikut di AWS CLI menunjukkan bidang khusus untuk kunci KMS di toko AWS CloudHSM kunci. Sebelum menjalankan perintah seperti ini, ganti contoh ID kunci KMS dengan nilai yang valid.

```
$ 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"
   }
}
```

Untuk bantuan menemukan kunci KMS di penyimpanan AWS CloudHSM kunci atau mengidentifikasi kunci di AWS CloudHSM klaster Anda yang berfungsi sebagai bahan utama untuk kunci KMS Anda, lihat. Menemukan kunci KMS dan bahan utama

Menggunakan kunci KMS di toko AWS CloudHSM kunci

Setelah Anda membuat kunci KMS enkripsi simetris di toko AWS CloudHSM kunci, Anda dapat menggunakannya untuk operasi kriptografi berikut:

- Enkripsi
- Dekripsi
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Operasi yang menghasilkan pasangan kunci data asimetris, GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintext, tidak didukung di penyimpanan kunci khusus.

Saat Anda menggunakan kunci KMS dalam permintaan, identifikasi kunci KMS dengan ID atau aliasnya; Anda tidak perlu menentukan penyimpanan AWS CloudHSM kunci atau klaster. AWS CloudHSM Respons mencakup bidang yang sama yang dikembalikan untuk kunci KMS enkripsi simetris apa pun.

Namun, ketika Anda menggunakan kunci KMS di toko AWS CloudHSM kunci, operasi kriptografi dilakukan sepenuhnya dalam AWS CloudHSM cluster yang terkait dengan penyimpanan AWS CloudHSM kunci. Operasi menggunakan materi kunci dalam cluster yang terkait dengan kunci KMS yang Anda pilih.

Untuk memungkinkan ini terjadi, syarat-syarat berikut diperlukan.

- Status kunci dari kunci KMS harus`Enabled`. Untuk menemukan status kunci, gunakan bidang Status di AWS KMSkonsol atau `KeyState` bidang dalam DescribeKeyrespons.
- Toko AWS CloudHSM kunci harus terhubung ke AWS CloudHSM klasternya. Statusnya di AWS KMSkonsol atau `ConnectionState` dalam DescribeCustomKeyStoresrespons harus`CONNECTED`.
- Klaster AWS CloudHSM yang terkait dengan penyimpanan kunci kustom harus berisi setidaknya satu HSM aktif. Untuk menemukan jumlah HSM aktif di cluster, gunakan AWS KMSkonsol, AWS CloudHSM konsol, atau DescribeClustersoperasi.
- AWS CloudHSMCluster harus berisi bahan kunci untuk kunci KMS. Jika material kunci dihapus dari klaster, atau HSM dibuat dari cadangan yang tidak menyertakan material kunci, operasi kriptografi akan gagal.

Jika kondisi ini tidak terpenuhi, operasi kriptografi gagal, dan AWS KMS mengembalikan pengecualian `KMSInvalidStateException`. Biasanya, Anda hanya perlu menghubungkan kembali toko AWS CloudHSM kunci. Untuk bantuan tambahan, lihat Cara memperbaiki kunci KMS yang gagal.

Saat menggunakan kunci KMS di toko AWS CloudHSM kunci, ketahuilah bahwa kunci KMS di setiap AWS CloudHSM toko kunci berbagi kuota permintaan toko kunci khusus untuk operasi kriptografi. Jika Anda melebihi kuota, AWS KMS mengembalikan `ThrottlingException`. Jika AWS CloudHSM cluster yang terkait dengan penyimpanan AWS CloudHSM kunci memproses banyak perintah, termasuk yang tidak terkait dengan penyimpanan AWS CloudHSM kunci, Anda mungkin mendapatkan `ThrottlingException` tingkat yang lebih rendah. Jika Anda mendapatkan `ThrottlingException` untuk permintaan apa pun, turunkan tingkat permintaan Anda dan coba lagi perintahnya. Untuk detail tentang kuota permintaan toko kunci kustom, lihatKuota permintaan toko kunci kustom.

Menemukan kunci KMS dan bahan utama

Jika Anda mengelola toko AWS CloudHSM kunci, Anda mungkin perlu mengidentifikasi kunci KMS di setiap toko AWS CloudHSM kunci. Misalnya, Anda mungkin perlu melakukan beberapa tugas berikut.

- Lacak kunci KMS di toko AWS CloudHSM kunci di AWS CloudTrail log.
- Memprediksi efek pada kunci KMS untuk memutuskan penyimpanan AWS CloudHSM kunci.
- Jadwalkan penghapusan kunci KMS sebelum Anda menghapus toko kunci. AWS CloudHSM

Selain itu, Anda mungkin ingin mengidentifikasi kunci di AWS CloudHSM cluster Anda yang berfungsi sebagai bahan utama untuk kunci KMS Anda. Meskipun AWS KMS mengelola kunci KMS dan materi utama, Anda masih mempertahankan kendali dan tanggung jawab untuk pengelolaan AWS CloudHSM cluster Anda, serta HSM dan cadangan dan kunci di HSM. Anda mungkin perlu mengidentifikasi kunci untuk mengaudit materi kunci, melindunginya dari penghapusan yang tidak disengaja, atau menghapusnya dari HSM dan cadangan cluster setelah menghapus kunci KMS.

Semua materi kunci untuk kunci KMS di toko AWS CloudHSM kunci Anda dimiliki oleh pengguna `kmsuser` kripto (CU). AWS KMSmenetapkan atribut label kunci, yang hanya dapat dilihat diAWS CloudHSM, ke Amazon Resource Name (ARN) dari kunci KMS.

Untuk menemukan kunci KMS dan bahan kunci, gunakan salah satu teknik berikut.

- Temukan kunci KMS di toko AWS CloudHSM kunci— Cara mengidentifikasi kunci KMS di satu atau semua toko AWS CloudHSM utama Anda.
- Temukan semua kunci untuk toko AWS CloudHSM kunci— Cara menemukan semua kunci di cluster Anda yang berfungsi sebagai bahan utama untuk kunci KMS di toko AWS CloudHSM kunci Anda.
- Temukan AWS CloudHSM kunci untuk kunci KMS— Cara menemukan kunci di cluster Anda yang berfungsi sebagai bahan utama untuk kunci KMS tertentu di toko AWS CloudHSM kunci Anda.
- Temukan kunci KMS untuk kunci AWS CloudHSM— Cara menemukan kunci KMS untuk kunci tertentu di cluster Anda.

Temukan kunci KMS di toko AWS CloudHSM kunci

Jika Anda mengelola toko AWS CloudHSM kunci, Anda mungkin perlu mengidentifikasi kunci KMS di setiap toko AWS CloudHSM kunci. Anda dapat menggunakan informasi ini melacak operasi kunci KMS di AWS CloudTrail log, memprediksi efek pada kunci KMS untuk memutuskan penyimpanan kunci kustom, atau menjadwalkan penghapusan kunci KMS sebelum Anda menghapus penyimpanan kunci. AWS CloudHSM

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci (konsol)

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci tertentu, pada halaman Kunci yang dikelola Pelanggan, lihat nilai di bidang Nama Toko Kunci Kustom atau ID Toko Kunci Kustom. Untuk mengidentifikasi kunci KMS di toko AWS CloudHSM kunci apa pun, cari kunci KMS dengan nilai Asal. AWS CloudHSM Untuk menambahkan kolom opsional ke layar, pilih ikon roda gigi di sudut kanan atas halaman.

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci (API)

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci, gunakan DescribeKeyoperasi ListKeysdan kemudian filter berdasarkan `CustomKeyStoreId` nilai. Sebelum menjalankan contoh, ganti nilai ID penyimpanan kunci kustom fiktif dengan nilai valid.

Bash

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci tertentu, dapatkan semua kunci KMS Anda di akun dan Wilayah. Kemudian filter dengan ID toko kunci khusus.

```
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
```

Untuk mendapatkan kunci KMS di toko AWS CloudHSM kunci apa pun di akun dan Wilayah, cari `CustomKeyStoreType` dengan nilai. `AWS_CloudHSM`

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

Untuk menemukan kunci KMS di toko AWS CloudHSM kunci tertentu, gunakan KmsKey cmdlet Get - KmsKeyList dan Get - untuk mendapatkan semua kunci KMS Anda di akun dan Wilayah. Kemudian filter dengan ID toko kunci khusus.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

Untuk mendapatkan kunci KMS di toko AWS CloudHSM kunci apa pun di akun dan Wilayah, filter untuk CustomKeyStoreType nilai. `AWS_CLOUDHSM`

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

Temukan semua kunci untuk toko AWS CloudHSM kunci

Anda dapat mengidentifikasi kunci di AWS CloudHSM cluster Anda yang berfungsi sebagai bahan utama untuk toko AWS CloudHSM kunci Anda. Untuk melakukan itu, gunakan findAllKeysperintah

di cloudhsm_mgmt_util untuk menemukan pegangan kunci dari semua kunci yang memiliki atau berbagi. `kmsuser` Kecuali Anda telah masuk sebagai `kmsuser` dan membuat kunci di luarAWS KMS, semua kunci yang `kmsuser` dimiliki mewakili materi kunci untuk kunci KMS.

Petugas crypto mana pun di cluster dapat menjalankan perintah ini tanpa memutuskan penyimpanan AWS CloudHSM kunci.

1. Mulai cloudhsm_mgmt_util dengan menggunakan prosedur yang dijelaskan dalam topik Memulai dengan CloudHSM Management Utility (CMU).
2. Masuk ke cloudhsm_mgmt_util menggunakan akun petugas kripto (CO).
3. Gunakan perintah listUsers untuk menemukan ID pengguna dari pengguna kripto `kmsuser`.

   Dalam contoh ini, `kmsuser` memiliki ID pengguna 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. Gunakan findAllKeysperintah untuk menemukan pegangan kunci dari semua kunci yang `kmsuser` memiliki atau berbagi. Ganti contoh ID pengguna (3) dengan ID pengguna aktual `kmsuser` di cluster Anda.

   Contoh output menunjukkan bahwa `kmsuser` memiliki kunci dengan nama kunci 8, 9, dan 262162 pada kedua HSM di klaster.

   ```
   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)
```

Temukan kunci KMS untuk kunci AWS CloudHSM

Jika Anda mengetahui pegangan kunci dari kunci yang `kmsuser` dimiliki di cluster, Anda dapat menggunakan label kunci untuk mengidentifikasi kunci KMS terkait di toko AWS CloudHSM kunci Anda.

Saat AWS KMS membuat materi kunci untuk kunci KMS di AWS CloudHSM cluster Anda, ia menulis Amazon Resource Name (ARN) dari kunci KMS di label kunci. Kecuali Anda telah mengubah nilai label, Anda dapat menggunakan perintah getAttribute di key_mgmt_util atau cloudhsm_mgmt_util untuk mengaitkan kunci dengan kunci KMS-nya.

Untuk menjalankan prosedur ini, Anda harus memutuskan sambungan penyimpanan AWS CloudHSM kunci sementara sehingga Anda dapat masuk sebagai `kmsuser` CU.

> **Note**
>
> Sementara toko kunci khusus terputus, semua upaya untuk membuat kunci KMS di toko kunci khusus atau menggunakan kunci KMS yang ada dalam operasi kriptografi akan gagal. Tindakan ini dapat mencegah pengguna menyimpan dan mengakses data sensitif.

1. Putuskan sambungan penyimpanan AWS CloudHSM kunci, jika belum terputus., lalu masuk ke key_mgmt_util as, seperti yang dijelaskan di. `kmsuser` Cara memutuskan koneksi dan login
2. Gunakan perintah `getAttribute` key_mgmt_util atau cloudhsm_mgmt_util untuk mendapatkan atribut label (`OBJ_ATTR_LABEL`, atribut `3`) untuk nama kunci tertentu.

   Sebagai contoh, perintah ini menggunakan `getAttribute` di cloudhsm_mgmt_util untuk mendapatkan atribut label (atribut `3`) dari kunci dengan nama kunci `262162`. Output menunjukkan bahwa kunci `262162` berfungsi sebagai bahan kunci untuk kunci KMS dengan ARN`arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`. Sebelum menjalankan perintah ini, ganti nama kunci contoh dengan yang valid.

Untuk daftar atribut kunci, gunakan perintah listAttributes atau lihat Referensi Atribut Kunci di Panduan Pengguna AWS CloudHSM.

```
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. Keluar dari key_mgmt_util atau cloudhsm_mgmt_util dan sambungkan kembali penyimpanan kunci seperti yang dijelaskan di. AWS CloudHSM Cara logout dan menghubungkan kembali

Temukan AWS CloudHSM kunci untuk kunci KMS

Anda dapat menggunakan ID kunci KMS dari kunci KMS di toko AWS CloudHSM kunci untuk mengidentifikasi kunci di AWS CloudHSM cluster Anda yang berfungsi sebagai materi utamanya. Kemudian Anda dapat menggunakan nama kunci untuk mengidentifikasi kunci di perintah klien AWS CloudHSM.

Saat AWS KMS membuat materi kunci untuk kunci KMS di AWS CloudHSM cluster Anda, ia menulis Amazon Resource Name (ARN) dari kunci KMS di label kunci. Kecuali Anda telah mengubah nilai label, Anda dapat menggunakan perintah FindKey di key_mgmt_util untuk mendapatkan pegangan kunci dari bahan kunci untuk kunci KMS. Untuk menjalankan prosedur ini, Anda harus memutuskan sambungan penyimpanan AWS CloudHSM kunci sementara sehingga Anda dapat masuk sebagai `kmsuser` CU.

> **Note**
>
> Sementara toko kunci khusus terputus, semua upaya untuk membuat kunci KMS di toko kunci khusus atau menggunakan kunci KMS yang ada dalam operasi kriptografi akan gagal. Tindakan ini dapat mencegah pengguna menyimpan dan mengakses data sensitif.

1. Putuskan sambungan penyimpanan AWS CloudHSM kunci, jika belum terputus, lalu masuk ke key_mgmt_util as, seperti yang dijelaskan di. `kmsuser` Cara memutuskan koneksi dan login
2. Gunakan perintah FindKey di key_mgmt_util untuk mencari kunci dengan label yang cocok dengan ARN kunci KMS di toko kunci Anda. AWS CloudHSM Ganti contoh ARN kunci KMS dalam nilai parameter (huruf kecil L untuk 'label') dengan ARN kunci KMS yang valid. `-l`

Misalnya, perintah ini menemukan kunci dengan label yang cocok dengan contoh kunci KMS `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab` ARN,. Output contoh menunjukkan bahwa kunci dengan pegangan kunci 262162 memiliki ARN kunci KMS yang ditentukan dalam labelnya. Anda sekarang dapat menggunakan nama kunci ini dalam perintah key_mgmt_util lainnya.

```
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. Keluar dari key_mgmt_util dan hubungkan kembali penyimpanan kunci kustom seperti yang dijelaskan dalam Cara logout dan menghubungkan kembali.

Penjadwalan penghapusan kunci KMS dari toko kunci AWS CloudHSM

Ketika Anda yakin bahwa Anda tidak perlu menggunakan untuk operasi kriptografi apa pun, Anda dapat menjadwalkan penghapusan kunci KMS. AWS KMS key Gunakan prosedur yang sama yang akan Anda gunakan untuk menjadwalkan penghapusan kunci KMS apa pun. AWS KMS Selain itu, jaga agar toko AWS CloudHSM kunci Anda tetap terhubung sehingga AWS KMS dapat menghapus materi kunci yang sesuai dari AWS CloudHSM klaster terkait saat masa tunggu berakhir.

Anda dapat memantau penjadwalan, pembatalan, dan penghapusan kunci KMS di log Anda. AWS CloudTrail

> **Warning**
>
> Menghapus kunci KMS adalah operasi yang merusak dan berpotensi berbahaya yang mencegah Anda memulihkan semua data yang dienkripsi di bawah kunci KMS. Sebelum menjadwalkan penghapusan kunci KMS, periksa penggunaan sebelumnya dari kunci KMS

dan buat CloudWatch alarm Amazon yang memberi tahu Anda ketika seseorang mencoba menggunakan kunci KMS saat sedang menunggu penghapusan. Jika memungkinkan, nonaktifkan kunci KMS, alih-alih menghapusnya.

Saat Anda menjadwalkan penghapusan kunci KMS dari toko kunci, status AWS CloudHSM kuncinya berubah menjadi penghapusan Tertunda. Kunci KMS tetap dalam status penghapusan Tertunda selama masa tunggu, bahkan jika kunci KMS menjadi tidak tersedia karena Anda telah memutuskan penyimpanan kunci kustom. Ini memungkinkan Anda untuk membatalkan penghapusan kunci KMS kapan saja selama masa tunggu.

Ketika masa tunggu berakhir, AWS KMS hapus kunci KMS dari. AWS KMS Kemudian AWS KMS membuat upaya terbaik untuk menghapus material kunci dari klaster AWS CloudHSM yang terkait. Jika AWS KMS tidak dapat menghapus material kunci, seperti ketika penyimpanan kunci terputus dari AWS KMS, Anda mungkin harus secara manual menghapus material kunci orphaned dari klaster.

AWS KMS tidak menghapus material kunci dari pencadangan klaster. Bahkan jika Anda menghapus kunci KMS dari AWS KMS dan menghapus materi kuncinya dari AWS CloudHSM klaster Anda, cluster yang dibuat dari cadangan mungkin berisi materi kunci yang dihapus. Untuk menghapus materi kunci secara permanen, lihat tanggal pembuatan kunci KMS. Kemudian hapus semua klaster pencadangan yang mungkin berisi material kunci.

Ketika Anda menjadwalkan penghapusan kunci KMS dari toko kunci, AWS CloudHSM kunci KMS menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhiLayanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Lihat perinciannya di Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Memecahkan masalah penyimpanan kunci kustom

AWS CloudHSMtoko-toko utama dirancang agar tersedia dan tangguh. Namun, ada beberapa kondisi kesalahan yang mungkin harus Anda perbaiki agar toko AWS CloudHSM kunci Anda tetap beroperasi.

Topik

- Cara memperbaiki kunci KMS yang tidak tersedia
- Cara memperbaiki kunci KMS yang gagal

- Cara memperbaiki kegagalan koneksi
- Bagaimana menanggapi kegagalan operasi kriptografi
- Cara memperbaiki kredensial kmsuser tidak valid
- Cara menghapus material kunci tanpa induk
- Bagaimana memulihkan materi kunci yang dihapus untuk kunci KMS
- Cara login sebagai kmsuser

Cara memperbaiki kunci KMS yang tidak tersedia

Keadaan kunci AWS KMS keys di toko AWS CloudHSM kunci biasanya`Enabled`. Seperti semua kunci KMS, status kunci berubah ketika Anda menonaktifkan kunci KMS di toko AWS CloudHSM kunci atau menjadwalkannya untuk dihapus. Namun, tidak seperti kunci KMS lainnya, kunci KMS di toko kunci khusus juga dapat memiliki status kunci. `Unavailable`

Status kunci `Unavailable` menunjukkan bahwa kunci KMS berada di penyimpanan kunci khusus yang sengaja terputus dan mencoba untuk menghubungkannya kembali, jika ada, gagal. Meskipun kunci KMS tidak tersedia, Anda dapat melihat dan mengelola kunci KMS, tetapi Anda tidak dapat menggunakannya untuk operasi kriptografi.

Untuk menemukan status kunci kunci KMS, pada halaman Kunci yang dikelola Pelanggan, lihat bidang Status kunci KMS. Atau, gunakan DescribeKeyoperasi dan lihat `KeyState` elemen dalam respons. Untuk detailnya, lihat Melihat kunci.

Kunci KMS di toko kunci kustom yang terputus akan memiliki status kunci atau. `Unavailable` `PendingDeletion` Kunci KMS yang dijadwalkan untuk dihapus dari toko kunci khusus memiliki status `Pending Deletion` kunci, bahkan ketika toko kunci khusus terputus. Hal ini memungkinkan Anda membatalkan penghapusan kunci terjadwal tanpa menghubungkan kembali penyimpanan kunci kustom.

Untuk memperbaiki kunci KMS yang tidak tersedia, sambungkan kembali toko kunci kustom. Setelah penyimpanan kunci kustom terhubung kembali, status kunci kunci KMS di toko kunci kustom secara otomatis dikembalikan ke keadaan sebelumnya, seperti `Enabled` atau. `Disabled` Kunci KMS yang tertunda penghapusan tetap berada di negara bagian. `PendingDeletion` Namun, sementara masalah tetap ada, mengaktifkan dan menonaktifkan kunci KMS yang tidak tersedia tidak mengubah status kuncinya. Tindakan mengaktifkan atau menonaktifkan hanya berlaku ketika kunci menjadi tersedia.

Untuk bantuan dengan koneksi yang gagal, lihat Cara memperbaiki kegagalan koneksi.

Cara memperbaiki kunci KMS yang gagal

Masalah dengan membuat dan menggunakan kunci KMS di toko-toko AWS CloudHSM utama dapat disebabkan oleh masalah dengan toko AWS CloudHSM kunci Anda, AWS CloudHSM cluster terkait, kunci KMS, atau materi utamanya.

Ketika penyimpanan AWS CloudHSM kunci terputus dari AWS CloudHSM klasternya, status kunci kunci KMS di toko kunci kustom adalah. `Unavailable` Semua permintaan untuk membuat kunci KMS di toko AWS CloudHSM kunci yang terputus mengembalikan pengecualian. `CustomKeyStoreInvalidStateException` Semua permintaan untuk mengenkripsi, mendekripsi, mengenkripsi ulang, atau menghasilkan kunci data mengembalikan pengecualian `KMSInvalidStateException`. Untuk memperbaiki masalah, sambungkan kembali toko AWS CloudHSM kunci.

Namun, upaya Anda untuk menggunakan kunci KMS di penyimpanan AWS CloudHSM kunci untuk operasi kriptografi mungkin gagal bahkan ketika status kuncinya `Enabled` dan status koneksi penyimpanan AWS CloudHSM kunci adalah. `Connected` Ini mungkin disebabkan oleh salah satu kondisi berikut.

- Materi kunci untuk kunci KMS mungkin telah dihapus dari AWS CloudHSM cluster terkait. Untuk menyelidiki, temukan pegangan kunci dari bahan kunci untuk kunci KMS dan, jika perlu, cobalah untuk memulihkan materi kunci.
- Semua HSM dihapus dari AWS CloudHSM cluster yang terkait dengan penyimpanan AWS CloudHSM kunci. Untuk menggunakan kunci KMS di penyimpanan AWS CloudHSM kunci dalam operasi kriptografi, AWS CloudHSM klaster harus berisi setidaknya satu HSM aktif. Untuk memverifikasi jumlah dan status HSM dalam sebuah AWS CloudHSM cluster, gunakan AWS CloudHSM konsol atau DescribeClustersoperasi. Untuk menambahkan HSM ke cluster, gunakan AWS CloudHSM konsol atau CreateHsmoperasi.
- AWS CloudHSMCluster yang terkait dengan penyimpanan AWS CloudHSM kunci telah dihapus. Untuk memperbaiki masalah, buat klaster dari cadangan yang berkaitan dengan klaster asli, seperti cadangan klaster asli, atau cadangan yang digunakan untuk membuat klaster asli. Kemudian, edit ID klaster dalam pengaturan penyimpanan kunci kustom. Untuk instruksi, lihat Bagaimana memulihkan materi kunci yang dihapus untuk kunci KMS.
- AWS CloudHSMCluster yang terkait dengan penyimpanan kunci khusus tidak memiliki sesi PKCS #11 yang tersedia. Ini biasanya terjadi selama periode lalu lintas burst tinggi ketika sesi tambahan diperlukan untuk melayani lalu lintas. Untuk menanggapi `KMSInternalException` dengan pesan kesalahan tentang sesi PKCS #11, mundur dan coba lagi permintaan tersebut.

Cara memperbaiki kegagalan koneksi

Jika Anda mencoba menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klasternya, tetapi operasi gagal, status koneksi penyimpanan AWS CloudHSM kunci berubah menjadi`FAILED`. Untuk menemukan status koneksi penyimpanan AWS CloudHSM kunci, gunakan AWS KMS konsol atau DescribeCustomKeyStoresoperasi.

Atau, beberapa upaya koneksi gagal dengan cepat karena kesalahan konfigurasi klaster dengan mudah terdeteksi. Dalam hal ini, status koneksi masih`DISCONNECTED`. Kegagalan ini mengembalikan pesan kesalahan atau pengecualian yang menjelaskan mengapa percobaan mengalami kegagalan. Tinjau deskripsi pengecualian dan persyaratan cluster, perbaiki masalah, perbarui toko AWS CloudHSM kunci, jika perlu, dan coba sambungkan lagi.

Ketika status koneksi`FAILED`, jalankan DescribeCustomKeyStoresoperasi dan lihat `ConnectionErrorCode` elemen dalam respons.

> **Note**
>
> Ketika status koneksi penyimpanan AWS CloudHSM kunci`FAILED`, Anda harus memutuskan penyimpanan AWS CloudHSM kunci sebelum mencoba menghubungkannya kembali. Anda tidak dapat menghubungkan toko AWS CloudHSM kunci dengan status `FAILED` koneksi.

- `CLUSTER_NOT_FOUND` menunjukkan bahwa AWS KMS tidak dapat menemukan klaster AWS CloudHSM dengan ID klaster yang ditentukan. Hal ini mungkin terjadi karena ID klaster yang salah disediakan untuk operasi API atau klaster telah dihapus dan tidak diganti. Untuk memperbaiki kesalahan ini, verifikasi ID cluster, seperti dengan menggunakan AWS CloudHSM konsol atau DescribeClustersoperasi. Jika klaster telah dihapus, buat klaster dari cadangan terbaru dari aslinya. Kemudian, lepaskan penyimpanan AWS CloudHSM kunci, edit pengaturan ID cluster penyimpanan AWS CloudHSM kunci, dan sambungkan kembali penyimpanan AWS CloudHSM kunci ke cluster.
- `INSUFFICIENT_CLOUDHSM_HSMS` menunjukkan bahwa klaster AWS CloudHSM terkait tidak berisi HSM apa pun. Untuk menghubungkan, klaster harus memiliki minimal satu HSM. Untuk menemukan jumlah HSM di cluster, gunakan DescribeClustersoperasi. Untuk mengatasi kesalahan ini, tambahkan minimal satu HSM ke klaster. Jika Anda menambahkan beberapa HSM, sebaiknya buat di Availability Zone yang berbeda.
- `INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET`menunjukkan bahwa tidak AWS KMS dapat menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM klasternya karena

setidaknya satu subnet pribadi yang terkait dengan cluster tidak memiliki alamat IP yang tersedia. Koneksi penyimpanan AWS CloudHSM kunci memerlukan satu alamat IP gratis di masing-masing subnet pribadi terkait, meskipun dua lebih disukai.

Anda tidak dapat menambahkan alamat IP (blok CIDR) ke subnet yang ada. Jika memungkinkan, pindahkan atau hapus sumber daya lain yang menggunakan alamat IP di subnet, seperti instans EC2 yang tidak digunakan atau antarmuka jaringan elastis. Jika tidak, Anda dapat membuat cluster dari cadangan klaster terbaru dengan subnet pribadi baru atau yang sudah ada yang memiliki lebih banyak ruang alamat kosong. AWS CloudHSM Kemudian, untuk mengaitkan cluster baru dengan penyimpanan AWS CloudHSM kunci Anda, lepaskan penyimpanan kunci kustom, ubah ID cluster penyimpanan AWS CloudHSM kunci ke ID cluster baru, dan coba sambungkan lagi.

> **Tip**
>
> Untuk menghindari menyetel ulang kata sandi `kmsuser`, gunakan cadangan terbaru dari klaster AWS CloudHSM.

- `INTERNAL_ERROR` menunjukkan bahwa AWS KMS tidak dapat menyelesaikan permintaan karena kesalahan internal. Coba lagi permintaannya. Untuk `ConnectCustomKeyStore` permintaan, putuskan sambungan penyimpanan AWS CloudHSM kunci sebelum mencoba menghubungkan lagi.
- `INVALID_CREDENTIALS` menunjukkan bahwa AWS KMS tidak dapat login ke klaster AWS CloudHSM terkait karena tidak memiliki kata sandi akun `kmsuser` yang benar. Untuk bantuan dengan kesalahan ini, lihat Cara memperbaiki kredensial `kmsuser` tidak valid.
- `NETWORK_ERRORS` biasanya menunjukkan masalah jaringan sementara. Putuskan sambungan toko AWS CloudHSM kunci, tunggu beberapa menit, dan coba sambungkan lagi.
- `SUBNET_NOT_FOUND` menunjukkan bahwa setidaknya satu subnet di konfigurasi klaster AWS CloudHSM telah dihapus. Jika AWS KMS tidak dapat menemukan semua subnet dalam konfigurasi cluster, upaya untuk menghubungkan penyimpanan AWS CloudHSM kunci ke AWS CloudHSM cluster gagal.

  Untuk memperbaiki kesalahan ini, buat klaster dari cadangan terbaru dari klaster AWS CloudHSM yang sama. (Proses ini membuat konfigurasi klaster baru dengan VPC dan subnet privat.) Pastikan klaster baru memenuhi persyaratan untuk penyimpanan kunci kustom, dan perhatikan ID klaster baru. Kemudian, untuk mengaitkan cluster baru dengan penyimpanan AWS CloudHSM kunci Anda, lepaskan penyimpanan kunci kustom, ubah ID cluster penyimpanan AWS CloudHSM kunci ke ID cluster baru, dan coba sambungkan lagi.

> **Tip**
>
> Untuk menghindari menyetel ulang kata sandi `kmsuser`, gunakan cadangan terbaru dari klaster AWS CloudHSM.

- `USER_LOCKED_OUT` menunjukkan bahwa akun pengguna kripto (CU) `kmsuser` dikunci dari klaster AWS CloudHSM yang terkait karena terlalu banyak upaya sandi yang gagal. Untuk bantuan dengan kesalahan ini, lihat Cara memperbaiki kredensial `kmsuser` tidak valid.

  Untuk memperbaiki kesalahan ini, lepaskan penyimpanan AWS CloudHSM kunci dan gunakan perintah ChangepSWD di cloudhsm_mgmt_util untuk mengubah kata sandi akun. `kmsuser` Kemudian, edit pengaturan kata sandi `kmsuser` untuk penyimpanan kunci kustom, dan coba untuk menyambungkan lagi. Untuk bantuan, gunakan prosedur yang dijelaskan dalam topik Cara memperbaiki kredensial `kmsuser` tidak valid.
- `USER_LOGGED_IN` menunjukkan bahwa akun CU `kmsuser` login ke klaster AWS CloudHSM yang terkait. Hal ini mencegah AWS KMS memutar kata sandi akun `kmsuser` dan login ke dalam klaster. Untuk memperbaiki kesalahan ini, logout CU `kmsuser` dari cluster. Jika Anda mengubah `kmsuser` kata sandi untuk masuk ke cluster, Anda juga harus memperbarui nilai kata sandi penyimpanan kunci untuk penyimpanan AWS CloudHSM kunci. Untuk bantuan, lihat Cara logout dan menghubungkan kembali.
- `USER_NOT_FOUND` menunjukkan bahwa AWS KMS tidak dapat menemukan akun CU `kmsuser` di klaster AWS CloudHSM yang terkait. Untuk memperbaiki kesalahan ini, buat akun `kmsuser` CU di cluster, lalu perbarui nilai kata sandi penyimpanan kunci untuk penyimpanan AWS CloudHSM kunci. Untuk bantuan, lihat Cara memperbaiki kredensial `kmsuser` tidak valid.

Bagaimana menanggapi kegagalan operasi kriptografi

Operasi kriptografi yang menggunakan kunci KMS di toko kunci khusus mungkin gagal dengan file. `KMSInvalidStateException` Pesan kesalahan berikut mungkin menyertai`KMSInvalidStateException`.

KMS tidak dapat berkomunikasi dengan klaster CloudHSM Anda. Ini mungkin masalah jaringan sementara. Jika Anda melihat kesalahan ini berulang kali, verifikasi bahwa ACL Jaringan dan aturan grup keamanan untuk VPC klaster AWS CloudHSM Anda sudah benar.

- Meskipun ini adalah kesalahan HTTPS 400, mungkin ini adalah hasil dari masalah jaringan sementara. Untuk menanggapi, mulailah dengan mencoba kembali permintaan. Namun, jika terus gagal, periksa konfigurasi komponen jaringan Anda. Kesalahan ini kemungkinan besar disebabkan oleh kesalahan konfigurasi komponen jaringan, seperti aturan firewall atau aturan grup keamanan VPC yang memblokir lalu lintas keluar.

KMS tidak dapat berkomunikasi dengan AWS CloudHSM cluster Anda karena kmsuser terkunci. Jika Anda melihat kesalahan ini berulang kali, lepaskan AWS CloudHSM kunci penyimpanan dan setel ulang kata sandi akun kmsuser. Perbarui kata sandi kmsuser untuk toko kunci khusus dan coba permintaan lagi.

- Pesan kesalahan ini menunjukkan bahwa akun pengguna `kmsuser` kripto (CU) dikunci dari AWS CloudHSM cluster terkait karena terlalu banyak upaya kata sandi yang gagal. Untuk bantuan dengan kesalahan ini, lihat Cara memutuskan koneksi dan login.

Cara memperbaiki kredensial **`kmsuser`** tidak valid

Saat Anda menghubungkan penyimpanan AWS CloudHSM kunci, AWS KMS masuk ke AWS CloudHSM klaster terkait sebagai pengguna `kmsuser` kripto (CU). Itu tetap masuk sampai toko AWS CloudHSM kunci terputus. Respons DescribeCustomKeyStores menunjukkan `ConnectionState` dari nilai `FAILED` dan `ConnectionErrorCode` dari `INVALID_CREDENTIALS`, seperti yang ditunjukkan dalam contoh berikut.

Jika Anda memutuskan penyimpanan AWS CloudHSM kunci dan mengubah `kmsuser` kata sandi, AWS KMS tidak dapat masuk ke AWS CloudHSM cluster dengan kredensyal akun CU. `kmsuser` Akibatnya, semua upaya untuk menghubungkan toko AWS CloudHSM kunci gagal. Respons `DescribeCustomKeyStores` menunjukkan `ConnectionState` dari nilai `FAILED` dan `ConnectionErrorCode` dari `INVALID_CREDENTIALS`, seperti yang ditunjukkan dalam contoh berikut.

```
$ 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"
   ],
}
```

Selain itu, setelah lima kali percobaan gagal untuk login ke klaster dengan kata sandi yang salah, AWS CloudHSM mengunci akun pengguna. Untuk login ke klaster, Anda harus mengubah kata sandi akun.

Jika AWS KMS mendapat respons penguncian saat mencoba masuk ke cluster sebagai `kmsuser` CU, permintaan untuk menghubungkan penyimpanan AWS CloudHSM kunci gagal. DescribeCustomKeyStoresRespons termasuk `ConnectionState` dari `FAILED` dan `ConnectionErrorCode` nilai`USER_LOCKED_OUT`, seperti yang ditunjukkan pada contoh berikut.

```
$ 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"
   ],
}
```

Untuk memperbaiki salah satu kondisi ini, gunakan prosedur berikut.

1. Putuskan sambungan toko AWS CloudHSM kunci.
2. Jalankan DescribeCustomKeyStoresoperasi dan lihat nilai `ConnectionErrorCode` elemen dalam respons.

   - Jika nilai `ConnectionErrorCode` adalah `INVALID_CREDENTIALS`, tentukan kata sandi saat ini untuk akun `kmsuser`. Jika perlu, gunakan perintah changePswd di cloudhsm_mgmt_util untuk mengatur kata sandi ke nilai yang diketahui.
   - Jika nilai `ConnectionErrorCode` adalah `USER_LOCKED_OUT`, Anda harus menggunakan perintah changePswd di cloudhsm_mgmt_util untuk mengubah kata sandi `kmsuser`.

3. Edit pengaturan kata sandi `kmsuser` sehingga cocok dengan kata sandi `kmsuser` saat ini dalam klaster. Tindakan ini memberi tahu kata sandi AWS KMS mana yang digunakan untuk login ke klaster. Itu tidak mengubah kata sandi `kmsuser` dalam klaster.
4. Hubungkan penyimpanan kunci kustom.

Cara menghapus material kunci tanpa induk

Setelah menjadwalkan penghapusan kunci KMS dari penyimpanan AWS CloudHSM kunci, Anda mungkin perlu menghapus materi kunci yang sesuai secara manual dari cluster terkait. AWS CloudHSM

Saat Anda membuat kunci KMS di penyimpanan AWS CloudHSM kunci, AWS KMS buat metadata kunci KMS AWS KMS dan buat materi kunci di cluster terkait. AWS CloudHSM Saat Anda menjadwalkan penghapusan kunci KMS di toko AWS CloudHSM kunci, setelah masa tunggu, AWS KMS menghapus metadata kunci KMS. Kemudian AWS KMS lakukan upaya terbaik untuk menghapus materi kunci yang sesuai dari AWS CloudHSM cluster. Upaya mungkin gagal jika AWS KMS tidak dapat mengakses klaster, seperti ketika terputus dari penyimpanan AWS CloudHSM kunci atau perubahan `kmsuser` kata sandi. AWS KMStidak mencoba menghapus materi kunci dari cadangan cluster.

AWS KMSmelaporkan hasil upayanya untuk menghapus materi kunci dari cluster dalam entri `DeleteKey` peristiwa AWS CloudTrail log Anda. Muncul dalam `backingKeysDeletionStatus` elemen `additionalEventData` elemen, seperti yang ditunjukkan pada entri contoh berikut. Entri ini juga mencakup ARN kunci KMS, AWS CloudHSM ID cluster, dan pegangan kunci dari material `backing-key-id` kunci ().


```
{
    "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"
}
```

Untuk menghapus material kunci dari klaster AWS CloudHSM yang terkait, gunakan prosedur seperti yang berikut. Contoh ini menggunakan alat baris perintah AWS CLI dan AWS CloudHSM, tetapi Anda dapat menggunakan AWS Management Console, bukan CLI.

1. Putuskan sambungan penyimpanan AWS CloudHSM kunci, jika belum terputus, lalu masuk ke key_mgmt_util, seperti yang dijelaskan di. Cara memutuskan koneksi dan login
2. Gunakan perintah deleteKey di key_mgmt_util untuk menghapus kunci dari HSM di klaster.

   Misalnya, perintah ini menghapus 262162 kunci dari HSM di klaster. Pegangan kunci tercantum dalam entri CloudTrail log.

   ```
   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. Keluar dari key_mgmt_util dan sambungkan kembali penyimpanan kunci seperti yang dijelaskan dalam. AWS CloudHSM Cara logout dan menghubungkan kembali

Bagaimana memulihkan materi kunci yang dihapus untuk kunci KMS

Jika materi kunci untuk sebuah AWS KMS key dihapus, kunci KMS tidak dapat digunakan dan semua ciphertext yang dienkripsi di bawah kunci KMS tidak dapat didekripsi. Hal ini dapat terjadi jika materi kunci untuk kunci KMS di penyimpanan AWS CloudHSM kunci dihapus dari AWS CloudHSM cluster terkait. Namun, itu mungkin untuk memulihkan material kunci.

Saat Anda membuat AWS KMS key (kunci KMS) di penyimpanan AWS CloudHSM kunci, AWS KMS log ke AWS CloudHSM cluster terkait dan membuat materi kunci untuk kunci KMS. Ini juga mengubah kata sandi ke nilai yang hanya diketahui dan tetap masuk selama penyimpanan AWS CloudHSM kunci terhubung. Karena hanya pemilik kunci, yaitu CU yang membuat kunci, yang dapat menghapus kuncinya, sehingga tidak mungkin kuncinya akan dihapus dari HSM secara tidak sengaja.

Namun, jika materi kunci untuk kunci KMS dihapus dari HSM di cluster, status kunci kunci KMS akhirnya berubah menjadi. `UNAVAILABLE` Jika Anda mencoba menggunakan kunci KMS untuk operasi kriptografi, operasi gagal dengan `KMSInvalidStateException` pengecualian. Yang terpenting, data apa pun yang dienkripsi di bawah kunci KMS tidak dapat didekripsi.

Dalam keadaan tertentu, Anda dapat memulihkan material kunci yang dihapus dengan membuat klaster dari cadangan yang berisi material utama. Strategi ini hanya berfungsi ketika setidaknya satu cadangan dibuat sementara kunci pernah ada dan sebelumnya dihapus.

Gunakan proses berikut untuk memulihkan material utama.

1. Temukan cadangan klaster yang berisi material kunci. Cadangan juga harus berisi semua pengguna dan kunci yang Anda butuhkan untuk mendukung klaster dan data terenkripsinya.

   Gunakan DescribeBackupsoperasi untuk membuat daftar cadangan untuk sebuah cluster. Kemudian gunakan stempel waktu cadangan untuk membantu Anda memilih cadangan. Untuk membatasi output ke cluster yang terkait dengan penyimpanan AWS CloudHSM kunci, gunakan `Filters` parameter, seperti yang ditunjukkan pada contoh berikut.

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
```

```
{
    "Backups": [
        {
            "ClusterId": "cluster-1a23b4cdefg",
            "BackupId": "backup-9g87f6edcba",
            "CreateTimestamp": 1536667238.328,
            "BackupState": "READY"
        },
             ...
    ]
}
```

2. Buat klaster dari cadangan yang dipilih. Verifikasi bahwa cadangan berisi kunci yang dihapus serta pengguna lain dan kunci yang diperlukan klaster.
3. Putuskan sambungan toko AWS CloudHSM kunci sehingga Anda dapat mengedit propertinya.
4. Edit ID cluster dari penyimpanan AWS CloudHSM kunci. Masukkan ID klaster dari klaster yang Anda buat dari cadangan. Karena klaster berbagi riwayat cadangan dengan klaster asli, ID klaster yang baru harus valid.
5. Hubungkan kembali toko AWS CloudHSM kunci.

Cara login sebagai **`kmsuser`**

Untuk membuat dan mengelola materi utama di AWS CloudHSM cluster untuk toko AWS CloudHSM kunci Anda, AWS KMS gunakan akun pengguna `kmsuser` kripto (CU). Anda membuat akun `kmsuser` CU di cluster Anda dan memberikan kata sandinya AWS KMS saat Anda membuat toko AWS CloudHSM kunci Anda.

Secara umum, AWS KMS mengelola akun `kmsuser`. Namun, untuk beberapa tugas, Anda perlu memutuskan penyimpanan AWS CloudHSM kunci, masuk ke cluster sebagai `kmsuser` CU, dan menggunakan alat baris perintah cloudhsm_mgmt_util dan key_mgmt_util.

> **Note**
>
> Sementara toko kunci khusus terputus, semua upaya untuk membuat kunci KMS di toko kunci khusus atau menggunakan kunci KMS yang ada dalam operasi kriptografi akan gagal. Tindakan ini dapat mencegah pengguna menyimpan dan mengakses data sensitif.

Topik ini menjelaskan cara memutuskan penyimpanan AWS CloudHSM kunci Anda dan masuk sebagaikmsuser, menjalankan alat baris AWS CloudHSM perintah, dan keluar dan menghubungkan kembali toko AWS CloudHSM kunci Anda.

Topik

- Cara memutuskan koneksi dan login
- Cara logout dan menghubungkan kembali

Cara memutuskan koneksi dan login

Gunakan prosedur berikut setiap kali Anda perlu login ke klaster yang terkait sebagai CU kmsuser.

1. Putuskan sambungan toko AWS CloudHSM kunci, jika belum terputus. Anda dapat menggunakan konsol AWS KMS atau API AWS KMS.

   Saat AWS CloudHSM kunci Anda AWS KMS terhubung, masuk sebagai filekmsuser. Hal ini mencegah Anda login sebagai kmsuser atau mengubah kata sandi kmsuser.

   Misalnya, perintah ini digunakan DisconnectCustomKeyStoreuntuk memutuskan sambungan penyimpanan kunci contoh. Ganti contoh ID penyimpanan AWS CloudHSM kunci dengan yang valid.

   ```
   $ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

2. Mulai cloudhsm_mgmt_util. Gunakan prosedur yang dijelaskan dalam bagian Bersiap untuk menjalankan cloudhsm_mgmt_util dari Panduan Pengguna AWS CloudHSM.
3. Login ke cloudhsm_mgmt_util pada klaster AWS CloudHSM klaster sebagai petugas kripto (CO).

   Misalnya, perintah ini login sebagai CO bernama admin. Ganti contoh nama pengguna dan kata sandi CO dengan nilai yang valid.

   ```
   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. Gunakan perintah ChangepSWD untuk mengubah kata sandi akun menjadi yang Anda kmsuser tahu. (AWS KMSmemutar kata sandi saat Anda menghubungkan toko AWS CloudHSM kunci

Anda.) Kata sandi harus berisi 7–32 karakter alfanumerik. Kata sandi peka huruf besar/kecil dan tidak dapat berisi karakter khusus.

Misalnya, perintah ini mengubah kata sandi `kmsuser` menjadi `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. Login ke key_mgmt_util atau cloudhsm_mgmt_util sebagai `kmsuser` menggunakan kata sandi yang Anda tetapkan. Untuk instruksi mendetail, lihat Memulai dengan cloudhsm_mgmt_util dan Memulai dengan key_mgmt_util. Alat yang Anda gunakan bergantung pada tugas Anda.

   Misalnya, perintah ini login ke 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
```

Cara logout dan menghubungkan kembali

1. Lakukan tugas, lalu logout dari alat baris perintah. Jika Anda tidak keluar, upaya untuk menghubungkan kembali toko AWS CloudHSM kunci Anda akan gagal.

```
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. Edit pengaturan kata sandi `kmsuser` untuk penyimpanan kunci kustom.

   Ini memberi tahu kata sandi AWS KMS saat ini untuk `kmsuser` di klaster. Jika Anda menghilangkan langkah ini, AWS KMS tidak akan dapat login ke klaster sebagai `kmsuser`, dan semua upaya untuk menyambungkan kembali penyimpanan kunci kustom Anda akan gagal. Anda dapat menggunakan AWS KMS konsol atau `KeyStorePassword` parameter UpdateCustomKeyStoreoperasi.

   Misalnya, perintah ini memberi tahu AWS KMS bahwa kata sandi saat ini adalah `tempPassword`. Ganti contoh kata sandi dengan kata sandi yang sebenarnya.

   ```
   $ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
   key-store-password tempPassword
   ```

3. Hubungkan kembali toko AWS KMS kunci ke AWS CloudHSM klasternya. Ganti contoh ID penyimpanan AWS CloudHSM kunci dengan yang valid. Selama proses koneksi, AWS KMS mengubah kata sandi `kmsuser` ke nilai yang hanya diketahui.

   ConnectCustomKeyStoreOperasi kembali dengan cepat, tetapi proses koneksi dapat memakan waktu lama. Respons awal tidak menunjukkan keberhasilan proses koneksi.

   ```
   $ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

4. Gunakan DescribeCustomKeyStoresoperasi untuk memverifikasi bahwa penyimpanan AWS CloudHSM kunci terhubung. Ganti contoh ID penyimpanan AWS CloudHSM kunci dengan yang valid.

   Dalam contoh ini, bidang status koneksi menunjukkan bahwa penyimpanan AWS CloudHSM kunci sekarang terhubung.

   ```
   $ 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"
   ],
}
```

Toko kunci eksternal

Penyimpanan kunci eksternal memungkinkan Anda untuk melindungi AWS sumber daya Anda menggunakan kunci kriptografi di luar. AWS Fitur canggih ini dirancang untuk beban kerja yang diatur yang harus Anda lindungi dengan kunci enkripsi yang disimpan dalam sistem manajemen kunci eksternal yang Anda kontrol. Toko kunci eksternal mendukung janji kedaulatan AWS digital untuk memberi Anda kontrol kedaulatan atas data Anda AWS, termasuk kemampuan untuk mengenkripsi dengan materi utama yang Anda miliki dan kendalikan di luar. AWS

Toko kunci eksternal adalah toko kunci khusus yang didukung oleh manajer kunci eksternal yang Anda miliki dan kelola di luar AWS. Manajer kunci eksternal Anda dapat berupa modul keamanan perangkat keras fisik atau virtual (HSM), atau sistem berbasis perangkat keras atau berbasis perangkat lunak apa pun yang mampu menghasilkan dan menggunakan kunci kriptografi. Operasi enkripsi dan dekripsi yang menggunakan kunci KMS di penyimpanan kunci eksternal dilakukan oleh manajer kunci eksternal Anda menggunakan bahan kunci kriptografi Anda, fitur yang dikenal sebagai hold your own keys (HyoKS).

AWS KMS tidak pernah berinteraksi langsung dengan pengelola kunci eksternal Anda, dan tidak dapat membuat, melihat, mengelola, atau menghapus kunci Anda. Sebaliknya, AWS KMS berinteraksi hanya dengan perangkat lunak proxy penyimpanan kunci eksternal (proxy XKS) yang Anda sediakan. Proxy penyimpanan kunci eksternal Anda memediasi semua komunikasi antara AWS KMS dan manajer kunci eksternal Anda. Ini mentransmisikan semua permintaan dari AWS KMS ke manajer kunci eksternal Anda, dan mengirimkan tanggapan dari manajer kunci eksternal Anda kembali ke. AWS KMS Proxy penyimpanan kunci eksternal juga menerjemahkan permintaan generik dari AWS KMS ke dalam format khusus vendor yang dapat dipahami oleh manajer kunci eksternal Anda, memungkinkan Anda untuk menggunakan toko kunci eksternal dengan manajer kunci dari berbagai vendor.

Anda dapat menggunakan kunci KMS di penyimpanan kunci eksternal untuk enkripsi sisi klien, termasuk dengan file. AWS Encryption SDK Tetapi penyimpanan kunci eksternal adalah sumber daya penting untuk enkripsi sisi server, memungkinkan Anda untuk melindungi AWS sumber daya Anda dalam beberapa Layanan AWS dengan kunci kriptografi Anda di luar. AWS Layanan AWS yang mendukung kunci yang dikelola pelanggan untuk enkripsi simetris juga mendukung kunci KMS di toko kunci eksternal. Untuk detail dukungan layanan, lihat Integrasi AWS Layanan.

Penyimpanan kunci eksternal memungkinkan Anda AWS KMS untuk menggunakan beban kerja yang diatur di mana kunci enkripsi harus disimpan dan digunakan di luar. AWS Tetapi mereka adalah penyimpangan besar dari model tanggung jawab bersama standar, dan membutuhkan beban operasional tambahan. Risiko yang lebih besar terhadap ketersediaan dan latensi akan, bagi sebagian besar pelanggan, melebihi manfaat keamanan yang dirasakan dari toko kunci eksternal.

Toko kunci eksternal memungkinkan Anda mengontrol akar kepercayaan. Data yang dienkripsi di bawah kunci KMS di toko kunci eksternal Anda dapat didekripsi hanya dengan menggunakan pengelola kunci eksternal yang Anda kontrol. Jika Anda mencabut sementara akses ke manajer kunci eksternal Anda, seperti dengan memutuskan penyimpanan kunci eksternal atau memutuskan koneksi manajer kunci eksternal Anda dari proxy penyimpanan kunci eksternal, AWS kehilangan semua akses ke kunci kriptografi Anda sampai Anda mengembalikannya. Selama interval itu, ciphertext yang dienkripsi di bawah kunci KMS Anda tidak dapat didekripsi. Jika Anda secara permanen mencabut akses ke pengelola kunci eksternal Anda, semua ciphertext yang dienkripsi di bawah kunci KMS di toko kunci eksternal Anda menjadi tidak dapat dipulihkan. Satu-satunya pengecualian adalah AWS layanan yang secara singkat menyimpan kunci data yang dilindungi oleh kunci KMS Anda. Kunci data ini terus berfungsi sampai Anda menonaktifkan sumber daya atau cache kedaluwarsa. Lihat perinciannya di Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Penyimpanan kunci eksternal membuka blokir beberapa kasus penggunaan untuk beban kerja yang diatur di mana kunci enkripsi harus tetap berada di bawah kendali Anda dan tidak dapat diakses. AWS Tetapi ini adalah perubahan besar dalam cara Anda mengoperasikan infrastruktur berbasis cloud dan perubahan signifikan dalam model tanggung jawab bersama. Untuk sebagian besar beban kerja, beban operasional tambahan dan risiko yang lebih besar terhadap ketersediaan dan kinerja akan melebihi manfaat keamanan yang dirasakan dari toko kunci eksternal.

Pelajari lebih lanjut:

- Mengumumkan Toko Kunci AWS KMS Eksternal di Blog AWS Berita.

Apakah saya memerlukan toko kunci eksternal?

Bagi sebagian besar pengguna, penyimpanan AWS KMS kunci default, yang dilindungi oleh modul keamanan hardaware tervalidasi FIPS 140-2 Security Level 3, memenuhi persyaratan keamanan, kontrol, dan peraturan mereka. Pengguna toko kunci eksternal dikenakan biaya yang besar, pemeliharaan, dan beban pemecahan masalah, dan risiko terhadap latensi, ketersediaan, dan keandalan.

Saat mempertimbangkan toko kunci eksternal, luangkan waktu untuk memahami alternatifnya, termasuk toko AWS CloudHSM kunci yang didukung oleh AWS CloudHSM cluster yang Anda miliki dan kelola, dan kunci KMS dengan materi kunci impor yang Anda hasilkan di HSM Anda sendiri dan dapat dihapus dari kunci KMS sesuai permintaan. Secara khusus, mengimpor bahan kunci dengan interval kedaluwarsa yang sangat singkat dapat memberikan tingkat kontrol yang sama tanpa risiko kinerja atau ketersediaan.

Penyimpanan kunci eksternal mungkin merupakan solusi yang tepat untuk organisasi Anda jika Anda memiliki persyaratan berikut:

- Anda diharuskan menggunakan kunci kriptografi di pengelola kunci lokal atau pengelola kunci di luar AWS yang Anda kendalikan.
- Anda harus menunjukkan bahwa kunci kriptografi Anda disimpan hanya di bawah kendali Anda di luar cloud.
- Anda harus mengenkripsi dan mendekripsi menggunakan kunci kriptografi dengan otorisasi independen.
- Material kunci harus tunduk pada jalur audit sekunder yang independen.

Jika Anda memilih penyimpanan kunci eksternal, batasi penggunaannya untuk beban kerja yang memerlukan perlindungan dengan kunci kriptografi di luar. AWS

Model tanggung jawab bersama

Kunci KMS standar menggunakan bahan kunci yang dihasilkan dan digunakan dalam HSM yang AWS KMS memiliki dan mengelola. Anda membuat kebijakan kontrol akses pada kunci KMS Anda dan mengonfigurasinya Layanan AWS menggunakan kunci KMS untuk melindungi sumber daya Anda. AWS KMS bertanggung jawab atas keamanan, ketersediaan, latensi, dan daya tahan material utama dalam kunci KMS Anda.

Kunci KMS di toko kunci eksternal bergantung pada materi utama dan operasi di manajer kunci eksternal Anda. Dengan demikian, keseimbangan tanggung jawab bergeser ke arah Anda. Anda bertanggung jawab atas keamanan, keandalan, daya tahan, dan kinerja kunci kriptografi di manajer kunci eksternal Anda. AWS KMS bertanggung jawab untuk segera menanggapi permintaan dan berkomunikasi dengan proxy penyimpanan kunci eksternal Anda, dan untuk menjaga standar keamanan kami. Untuk memastikan bahwa setiap kunci eksternal menyimpan ciphertext setidaknya sekuat AWS KMS ciphertext standar, AWS KMS pertama-tama mengenkripsi semua plaintext

dengan materi AWS KMS kunci khusus untuk kunci KMS Anda, dan kemudian mengirimkannya ke pengelola kunci eksternal Anda untuk enkripsi dengan kunci eksternal Anda, prosedur yang dikenal sebagai enkripsi ganda. Akibatnya, baik AWS KMS maupun pemilik materi kunci eksternal tidak dapat mendekripsi ciphertext terenkripsi ganda saja.

Anda bertanggung jawab untuk memelihara manajer kunci eksternal yang memenuhi standar peraturan dan kinerja Anda, untuk memasok dan memelihara proxy penyimpanan kunci eksternal yang sesuai dengan Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal, dan untuk memastikan ketersediaan dan daya tahan materi utama Anda. Anda juga harus membuat, mengkonfigurasi, dan memelihara penyimpanan kunci eksternal. Ketika kesalahan muncul yang disebabkan oleh komponen yang Anda pertahankan, Anda harus siap untuk mengidentifikasi dan menyelesaikan kesalahan sehingga AWS layanan dapat mengakses sumber daya Anda tanpa gangguan yang tidak semestinya. AWS KMS memberikan panduan pemecahan masalah untuk membantu Anda menentukan penyebab masalah dan resolusi yang paling mungkin.

Tinjau CloudWatch metrik dan dimensi Amazon yang AWS KMS mencatat untuk penyimpanan kunci eksternal. AWS KMS sangat menyarankan agar Anda membuat CloudWatch alarm untuk memantau penyimpanan kunci eksternal Anda sehingga Anda dapat mendeteksi tanda-tanda awal masalah kinerja dan operasional sebelum terjadi.

Apa yang berubah?

Toko kunci eksternal hanya mendukung kunci KMS enkripsi simetris. Di dalam AWS KMS, Anda menggunakan dan mengelola kunci KMS di penyimpanan kunci eksternal dengan cara yang sama seperti Anda mengelola kunci yang dikelola pelanggan lainnya, termasuk pengaturan kebijakan kontrol akses dan pemantauan penggunaan kunci. Anda menggunakan API yang sama dengan parameter yang sama untuk meminta operasi kriptografi dengan kunci KMS di penyimpanan kunci eksternal yang Anda gunakan untuk kunci KMS apa pun. Harga juga sama dengan kunci KMS standar. Untuk detailnyaMengelola kunci KMS di toko kunci eksternal, lihatMenggunakan kunci KMS di toko kunci eksternal, dan AWS Key Management Service Harga.

Namun, dengan penyimpanan kunci eksternal, prinsip-prinsip berikut berubah:

- Anda bertanggung jawab atas ketersediaan, daya tahan, dan latensi operasi utama.
- Anda bertanggung jawab atas semua biaya untuk mengembangkan, membeli, mengoperasikan, dan melisensikan sistem manajer kunci eksternal Anda.
- Anda dapat menerapkan otorisasi independen dari semua permintaan dari AWS KMS proxy penyimpanan kunci eksternal Anda.

- Anda dapat memantau, mengaudit, dan mencatat semua operasi proxy penyimpanan kunci eksternal Anda, dan semua operasi manajer kunci eksternal Anda yang terkait dengan AWS KMS permintaan.

Di mana saya mulai?

Untuk membuat dan mengelola penyimpanan kunci eksternal, Anda harus memilih opsi konektivitas proxy penyimpanan kunci eksternal, merakit prasyarat, dan membuat dan mengonfigurasi penyimpanan kunci eksternal Anda. Untuk memulai, lihatMerencanakan toko kunci eksternal.

Kuota

AWS KMS memungkinkan hingga 10 toko kunci khusus di masing-masing Akun AWS dan Wilayah, termasuk toko utama dan toko AWS CloudHSM kunci eksternal, terlepas dari status koneksinya. Selain itu, ada kuota AWS KMS permintaan tentang penggunaan kunci KMS di toko kunci eksternal.

Jika Anda memilih konektivitas proxy VPC untuk proxy penyimpanan kunci eksternal Anda, mungkin juga ada kuota pada komponen yang diperlukan, seperti VPC, subnet, dan penyeimbang beban jaringan. Untuk informasi tentang kuota ini, gunakan konsol Service Quotas.

Daerah

Untuk meminimalkan latensi jaringan, buat komponen penyimpanan kunci eksternal Anda yang Wilayah AWS paling dekat dengan pengelola kunci eksternal Anda. Jika memungkinkan, pilih Wilayah dengan waktu pulang-pergi jaringan (RTT) 35 milidetik atau kurang.

Toko kunci eksternal didukung Wilayah AWS di semua yang AWS KMS didukung kecuali untuk China (Beijing) dan China (Ningxia).

Fitur yang tidak didukung

AWS KMS tidak mendukung fitur berikut di toko kunci khusus.

- Kunci Asymmetric KMS
- Pasangan kunci data asimetris
- Kunci HMAC KMS
- Kunci KMS dengan bahan kunci impor

- Rotasi kunci otomatis
- Kunci Multi-Region

Topik

- Konsep toko kunci eksternal
- Cara kerja toko kunci eksternal
- Mengontrol akses ke toko kunci eksternal Anda
- Merencanakan toko kunci eksternal
- Mengelola toko kunci eksternal
- Mengelola kunci KMS di toko kunci eksternal
- Memecahkan masalah toko kunci eksternal

Konsep toko kunci eksternal

Topik ini menjelaskan beberapa konsep yang digunakan di toko kunci eksternal.

Topik

- Toko kunci eksternal
- Manajer kunci eksternal
- Kunci eksternal
- Proksi penyimpanan kunci eksternal
- Konektivitas proxy penyimpanan kunci eksternal
- Kredensi otentikasi proxy penyimpanan kunci eksternal
- API Proxy
- Enkripsi ganda

Toko kunci eksternal

Toko kunci eksternal adalah toko kunci AWS KMS khusus yang didukung oleh manajer kunci eksternal di luar AWS yang Anda miliki dan kelola. Setiap kunci KMS di penyimpanan kunci eksternal dikaitkan dengan kunci eksternal di manajer kunci eksternal Anda. Ketika Anda menggunakan kunci KMS di penyimpanan kunci eksternal untuk enkripsi atau dekripsi, operasi dilakukan di manajer kunci eksternal Anda menggunakan kunci eksternal Anda, pengaturan yang dikenal sebagai Tahan Kunci

Anda Sendiri (HYOK). Fitur ini dirancang untuk organisasi yang diperlukan untuk mempertahankan kunci kriptografi mereka di manajer kunci eksternal mereka sendiri.

Penyimpanan kunci eksternal memastikan bahwa kunci kriptografi dan operasi yang melindungi AWS sumber daya Anda tetap berada di manajer kunci eksternal Anda di bawah kendali Anda. AWS KMS mengirimkan permintaan ke pengelola kunci eksternal Anda untuk mengenkripsi dan mendekripsi data, tetapi AWS KMS tidak dapat membuat, menghapus, atau mengelola kunci eksternal apa pun. Semua permintaan dari AWS KMS manajer kunci eksternal Anda dimediasi oleh komponen perangkat lunak proxy penyimpanan kunci eksternal yang Anda suplai, miliki, dan kelola.

AWS layanan yang mendukung kunci yang dikelola AWS KMS pelanggan dapat menggunakan kunci KMS di toko kunci eksternal Anda untuk melindungi data Anda. Akibatnya, data Anda pada akhirnya dilindungi oleh kunci Anda menggunakan operasi enkripsi Anda di pengelola kunci eksternal Anda.

Kunci KMS di toko kunci eksternal memiliki model kepercayaan yang berbeda secara fundamental, pengaturan tanggung jawab bersama, dan harapan kinerja daripada kunci KMS standar. Dengan toko kunci eksternal, Anda bertanggung jawab atas keamanan dan integritas materi utama dan operasi kriptografi. Ketersediaan dan latensi kunci KMS di toko kunci eksternal dipengaruhi oleh perangkat keras, perangkat lunak, komponen jaringan, dan jarak antara AWS KMS dan manajer kunci eksternal Anda. Anda juga cenderung mengeluarkan biaya tambahan untuk manajer kunci eksternal Anda dan untuk infrastruktur jaringan dan load balancing yang Anda butuhkan untuk manajer kunci eksternal Anda untuk berkomunikasi dengan AWS KMS

Anda dapat menggunakan penyimpanan kunci eksternal sebagai bagian dari strategi perlindungan data yang lebih luas. Untuk setiap AWS sumber daya yang Anda lindungi, Anda dapat memutuskan mana yang memerlukan kunci KMS di toko kunci eksternal dan mana yang dapat dilindungi oleh kunci KMS standar. Ini memberi Anda fleksibilitas untuk memilih kunci KMS untuk klasifikasi data, aplikasi, atau proyek tertentu.

Manajer kunci eksternal

Manajer kunci eksternal adalah komponen di luar AWS yang dapat menghasilkan kunci simetris AES 256-bit dan melakukan enkripsi dan dekripsi simetris. Manajer kunci eksternal untuk penyimpanan kunci eksternal dapat berupa modul keamanan perangkat keras fisik (HSM), HSM virtual, atau manajer kunci perangkat lunak dengan atau tanpa komponen HSM. Ini dapat ditemukan di mana saja di luar AWS, termasuk di tempat Anda, di pusat data lokal atau jarak jauh, atau di cloud apa pun. Penyimpanan kunci eksternal Anda dapat didukung oleh satu manajer kunci eksternal atau beberapa contoh manajer kunci terkait yang berbagi kunci kriptografi, seperti klaster HSM. Toko kunci eksternal

dirancang untuk mendukung berbagai manajer eksternal dari vendor yang berbeda. Untuk detail tentang persyaratan untuk manajer kunci eksternal Anda, lihatMerencanakan toko kunci eksternal.

Kunci eksternal

Setiap kunci KMS di toko kunci eksternal dikaitkan dengan kunci kriptografi di manajer kunci eksternal Anda yang dikenal sebagai kunci eksternal. Ketika Anda mengenkripsi atau mendekripsi dengan kunci KMS di toko kunci eksternal Anda, operasi kriptografi dilakukan di manajer kunci eksternal Anda menggunakan kunci eksternal Anda.

> **Warning**
>
> Kunci eksternal sangat penting untuk pengoperasian kunci KMS. Jika kunci eksternal hilang atau dihapus, ciphertext yang dienkripsi di bawah kunci KMS terkait tidak dapat dipulihkan.

Untuk penyimpanan kunci eksternal, kunci eksternal harus berupa kunci AES 256-bit yang diaktifkan dan dapat melakukan enkripsi dan dekripsi. Untuk persyaratan kunci eksternal yang terperinci, lihatPersyaratan untuk kunci KMS di toko kunci eksternal.

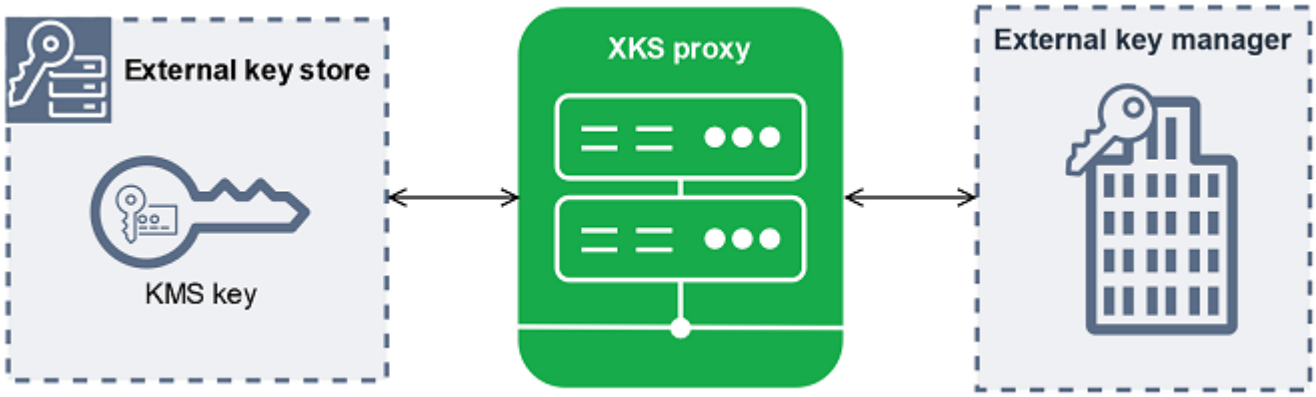
AWS KMS tidak dapat membuat, menghapus, atau mengelola kunci eksternal apa pun. Materi kunci kriptografi Anda tidak pernah meninggalkan manajer kunci eksternal Anda.Saat Anda membuat kunci KMS di penyimpanan kunci eksternal, Anda memberikan ID kunci eksternal (). `XksKeyId` Anda tidak dapat mengubah ID kunci eksternal yang terkait dengan kunci KMS, meskipun pengelola kunci eksternal Anda dapat memutar materi kunci yang terkait dengan ID kunci eksternal.

Selain kunci eksternal Anda, kunci KMS di toko kunci eksternal juga memiliki materi AWS KMS utama. Data yang dilindungi oleh kunci KMS dienkripsi terlebih dahulu dengan AWS KMS menggunakan materi AWS KMS kunci, dan kemudian oleh pengelola kunci eksternal Anda menggunakan kunci eksternal Anda. Proses enkripsi ganda ini memastikan bahwa ciphertext yang dilindungi oleh kunci KMS Anda selalu setidaknya sekuat ciphertext yang hanya dilindungi oleh. AWS KMS

Banyak kunci kriptografi memiliki berbagai jenis pengidentifikasi. Saat membuat kunci KMS di penyimpanan kunci eksternal, berikan ID kunci eksternal yang digunakan proxy penyimpanan kunci eksternal untuk merujuk ke kunci eksternal. Jika Anda menggunakan pengenal yang salah, upaya Anda untuk membuat kunci KMS di penyimpanan kunci eksternal Anda gagal.

Proksi penyimpanan kunci eksternal

Proxy penyimpanan kunci eksternal (“XKS proxy”) adalah aplikasi perangkat lunak milik pelanggan dan dikelola pelanggan yang memediasi semua komunikasi antara AWS KMS dan manajer kunci eksternal Anda. Ini juga menerjemahkan AWS KMS permintaan generik ke dalam format yang dipahami manajer kunci eksternal khusus vendor Anda. Proxy penyimpanan kunci eksternal diperlukan untuk penyimpanan kunci eksternal. Setiap penyimpanan kunci eksternal dikaitkan dengan satu proxy penyimpanan kunci eksternal.



AWS KMS tidak dapat membuat, menghapus, atau mengelola kunci eksternal apa pun. Materi kunci kriptografi Anda tidak pernah meninggalkan manajer kunci eksternal Anda. Semua komunikasi antara AWS KMS dan manajer kunci eksternal Anda dimediasi oleh proxy penyimpanan kunci eksternal Anda. AWS KMS mengirimkan permintaan ke proxy penyimpanan kunci eksternal dan menerima tanggapan dari proxy penyimpanan kunci eksternal. Proxy penyimpanan kunci eksternal bertanggung jawab untuk mengirimkan permintaan dari AWS KMS manajer kunci eksternal Anda dan mengirimkan tanggapan dari manajer kunci eksternal Anda kembali ke AWS KMS

Anda memiliki dan mengelola proxy penyimpanan kunci eksternal untuk toko kunci eksternal Anda, dan Anda bertanggung jawab atas pemeliharaan dan operasinya. Anda dapat mengembangkan proxy penyimpanan kunci eksternal berdasarkan spesifikasi API proxy toko kunci eksternal sumber terbuka yang AWS KMS menerbitkan atau membeli aplikasi proxy dari vendor. Proxy penyimpanan kunci eksternal Anda mungkin disertakan dalam pengelola kunci eksternal Anda. Untuk mendukung pengembangan proxy, AWS KMS juga menyediakan contoh proxy penyimpanan kunci eksternal (aws-kms-xks-proxy) dan klien pengujian (xks-kms-xksproxy-test-client) yang memverifikasi bahwa proxy penyimpanan kunci eksternal Anda sesuai dengan spesifikasi.

Untuk mengautentikasi AWS KMS, proxy menggunakan sertifikat TLS sisi server. Untuk mengautentikasi ke proxy Anda, AWS KMS tandatangani semua permintaan ke proxy penyimpanan kunci eksternal Anda dengan kredensi otentikasi proxy SiGv4. Secara opsional, proxy Anda

dapat mengaktifkan TLS bersama (mTL) untuk jaminan tambahan bahwa proxy hanya menerima permintaan dari. AWS KMS

Proxy penyimpanan kunci eksternal Anda harus mendukung HTTP/1.1 atau yang lebih baru dan TLS 1.2 atau yang lebih baru dengan setidaknya satu dari rangkaian sandi berikut:

- TLS_AES_256_GCM_SHA384 (TLS 1.3)
- TLS_CHACHA20_POLY1305_SHA256 (TLS 1.3)

> **Note**
>
> AWS GovCloud (US) Region Tidak mendukung TLS_CHACHA20_POLY1305_SHA256.

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)

Untuk membuat dan menggunakan kunci KMS di toko kunci eksternal Anda, Anda harus terlebih dahulu menghubungkan toko kunci eksternal ke proxy penyimpanan kunci eksternal. Anda juga dapat memutuskan penyimpanan kunci eksternal Anda dari proksi sesuai permintaan. Ketika Anda melakukannya, semua kunci KMS di penyimpanan kunci eksternal menjadi tidak tersedia; mereka tidak dapat digunakan dalam operasi kriptografi apa pun.

Konektivitas proxy penyimpanan kunci eksternal

Konektivitas proxy penyimpanan kunci eksternal (“konektivitas proxy XKS”) menjelaskan metode yang AWS KMS digunakan untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda.

Anda menentukan opsi konektivitas proxy saat membuat penyimpanan kunci eksternal, dan itu menjadi properti penyimpanan kunci eksternal. Anda dapat mengubah opsi konektivitas proxy Anda dengan memperbarui properti penyimpanan kunci kustom, tetapi Anda harus yakin bahwa proxy penyimpanan kunci eksternal Anda masih dapat mengakses kunci eksternal yang sama.

AWS KMS mendukung opsi konektivitas berikut:

- Konektivitas titik akhir publik — AWS KMS mengirimkan permintaan untuk proxy penyimpanan kunci eksternal Anda melalui internet ke titik akhir publik yang Anda kontrol. Opsi ini mudah dibuat dan dipelihara, tetapi mungkin tidak memenuhi persyaratan keamanan untuk setiap instalasi.
- Konektivitas layanan titik akhir VPC — AWS KMS mengirimkan permintaan ke layanan endpoint Amazon Virtual Private Cloud (Amazon VPC) yang Anda buat dan pelihara. Anda dapat meng-

host proxy penyimpanan kunci eksternal Anda di dalam VPC Amazon Anda, atau meng-host proxy penyimpanan kunci eksternal Anda di luar AWS dan menggunakan VPC Amazon hanya untuk komunikasi.

Untuk detail tentang opsi konektivitas proxy penyimpanan kunci eksternal, lihatMemilih opsi konektivitas proxy.

Kredensi otentikasi proxy penyimpanan kunci eksternal

Untuk mengautentikasi ke proxy penyimpanan kunci eksternal Anda, AWS KMS tandatangani semua permintaan ke proxy penyimpanan kunci eksternal Anda dengan kredensi otentikasi Signature V4 (SigV4). Anda membuat dan mempertahankan kredensi otentikasi pada proxy Anda, kemudian memberikan kredensi ini AWS KMS ketika Anda membuat toko eksternal Anda.

> **Note**
>
> Kredensyal SiGv4 yang AWS KMS digunakan untuk menandatangani permintaan ke proxy XKS tidak terkait dengan kredensyal SigV4 apa pun yang terkait dengan prinsipal di Anda. AWS Identity and Access Management Akun AWS Jangan gunakan kembali kredensyal IAM SiGv4 apa pun untuk proxy penyimpanan kunci eksternal Anda.

Setiap kredensi otentikasi proxy memiliki dua bagian. Anda harus menyediakan kedua bagian saat membuat penyimpanan kunci eksternal atau memperbarui kredensi otentikasi untuk penyimpanan kunci eksternal Anda.

- ID kunci akses: Mengidentifikasi kunci akses rahasia. Anda dapat memberikan ID ini dalam teks biasa.
- Kunci akses rahasia: Bagian rahasia dari kredensi. AWS KMS mengenkripsi kunci akses rahasia di kredensi sebelum menyimpannya.

Anda dapat mengedit setelan kredensyal kapan saja, seperti ketika Anda memasukkan nilai yang salah, ketika Anda mengubah kredensi Anda pada proxy, atau ketika proxy Anda memutar kredensialnya. Untuk detail teknis tentang AWS KMS autentikasi ke proxy penyimpanan kunci eksternal, lihat Otentikasi di Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal.

Untuk memungkinkan Anda memutar kredensi Anda tanpa mengganggu Layanan AWS yang menggunakan kunci KMS di penyimpanan kunci eksternal Anda, kami menyarankan agar proxy

penyimpanan kunci eksternal mendukung setidaknya dua kredensyal otentikasi yang valid untuk. AWS KMS Ini memastikan bahwa kredensi Anda sebelumnya terus berfungsi saat Anda memberikan kredensi baru Anda. AWS KMS

Untuk membantu Anda melacak usia kredensi autentikasi proxy Anda, AWS KMS tentukan metrik Amazon CloudWatch ,. XksProxyCredentialAge Anda dapat menggunakan metrik ini untuk membuat CloudWatch alarm yang memberi tahu Anda ketika usia kredensi Anda mencapai ambang batas yang Anda tetapkan.

Untuk memberikan jaminan tambahan bahwa proxy penyimpanan kunci eksternal Anda hanya merespons AWS KMS, beberapa proxy kunci eksternal mendukung mutual Transport Layer Security (mTLS). Lihat perinciannya di otentikasi mTLS (opsional).

API Proxy

Untuk mendukung penyimpanan kunci AWS KMS eksternal, proxy penyimpanan kunci eksternal harus mengimplementasikan API proxy yang diperlukan seperti yang dijelaskan dalam Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal. Permintaan API proxy ini adalah satu-satunya permintaan yang AWS KMS dikirim ke proxy. Meskipun Anda tidak pernah mengirim permintaan ini secara langsung, mengetahuinya dapat membantu Anda memperbaiki masalah apa pun yang mungkin timbul dengan penyimpanan kunci eksternal atau proksi. Misalnya, AWS KMS sertakan informasi tentang latensi dan tingkat keberhasilan panggilan API ini dalam CloudWatch metrik Amazon untuk penyimpanan kunci eksternal. Lihat perinciannya di Memantau toko kunci eksternal.

Tabel berikut mencantumkan dan menjelaskan masing-masing API proxy. Ini juga mencakup AWS KMS operasi yang memicu panggilan ke API proxy dan pengecualian AWS KMS operasi apa pun yang terkait dengan API proxy.

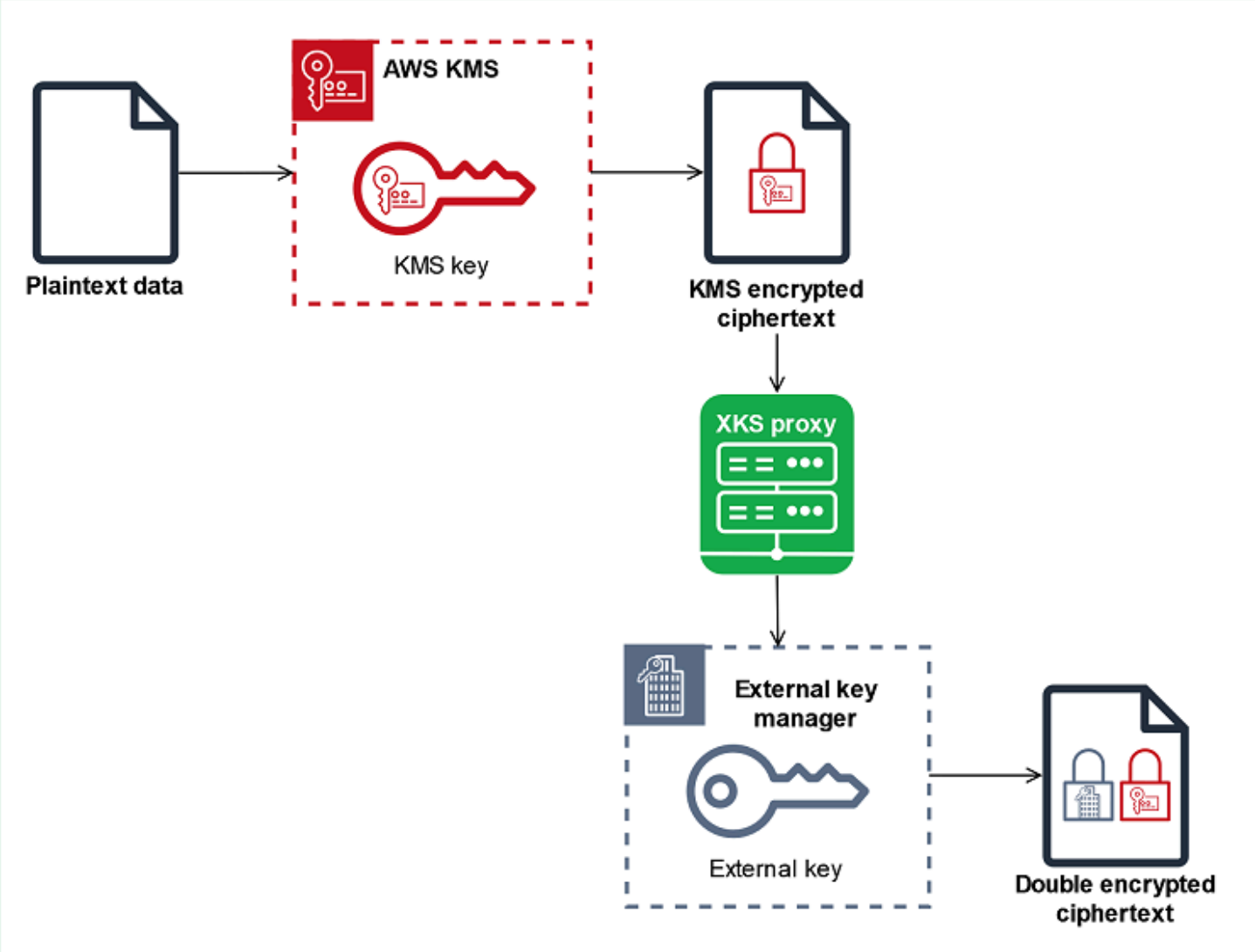
API Proksi	Deskripsi	AWS KMS Operasi terkait
Dekripsi	AWS KMS mengirimkan ciphertext untuk didekripsi, dan ID kunci eksternal untuk digunakan. Algoritma enkripsi yang diperlukan adalah AES_GCM.	Dekripsi, ReEncrypt
Enkripsi	AWS KMS mengirimkan data yang akan dienkripsi, dan ID kunci eksternal	Enkripsi,, GenerateDataKey, GenerateDataKeyWithoutPlaintextReEncrypt

API Proksi	Deskripsi	AWS KMS Operasi terkait
	untuk digunakan. Algoritma enkripsi yang diperlukan adalah AES_GCM.	
GetHealth Status	AWS KMS meminta informasi tentang status proxy dan manajer kunci eksternal Anda. Status setiap manajer kunci eksternal dapat menjadi salah satu dari berikut ini. • `Active`: Sehat; dapat melayani lalu lintas • `Degraded`: Tidak sehat, tetapi dapat melayani lalu lintas • `Unavailable` : Tidak sehat; tidak dapat melayani lalu lintas	CreateCustomKeyStore(untuk konektivitas titik akhir publik), ConnectCustomKeyStore(untuk konektivitas layanan titik akhir VPC) Jika semua instance pengelola kunci eksternal`Unavailable` , upaya untuk membuat atau menghubungkan penyimpanan kunci gagal. `XksProxyUriUnreachableException`
GetKeyMetadata	AWS KMS meminta informasi tentang kunci eksternal yang terkait dengan kunci KMS di toko kunci eksternal Anda. Responsnya mencakup spesifikasi kunci (`AES_256`), penggunaan kunci (`[ENCRYPT, DECRYPT]`), dan apakah kunci eksternal adalah `ENABLED` atau`DISABLED`.	CreateKey Jika spesifikasi kunci tidak`AES_256`, atau penggunaan kunci tidak`[ENCRYPT, DECRYPT]`, atau statusnya`DISABLED`, `CreateKey` operasi gagal dengan`XksKeyInvalidConfigurationException` .

Enkripsi ganda

Data yang dienkripsi oleh kunci KMS di toko kunci eksternal dienkripsi dua kali. Pertama, AWS KMS mengenkripsi data dengan bahan AWS KMS kunci khusus untuk kunci KMS. Kemudian ciphertext AWS KMS-enkripsi dienkripsi oleh pengelola kunci eksternal Anda menggunakan kunci eksternal Anda. Proses ini dikenal sebagai enkripsi ganda.

Enkripsi ganda memastikan bahwa data yang dienkripsi oleh kunci KMS di penyimpanan kunci eksternal setidaknya sekuat ciphertext yang dienkripsi oleh kunci KMS standar. Ini juga melindungi plaintext Anda dalam perjalanan dari AWS KMS ke proxy toko kunci eksternal Anda. Dengan enkripsi ganda, Anda mempertahankan kendali penuh atas ciphertext Anda. Jika Anda secara permanen mencabut AWS akses ke kunci eksternal Anda melalui proxy eksternal Anda, setiap ciphertext yang tersisa di dalamnya AWS secara efektif diparut kripto.



Untuk mengaktifkan enkripsi ganda, setiap kunci KMS di toko kunci eksternal memiliki dua kunci pendukung kriptografi:

- Bahan AWS KMS kunci yang unik untuk kunci KMS. Bahan utama ini dihasilkan dan hanya digunakan dalam modul keamanan hardaware bersertifikat AWS KMS FIPS 140-2 Security Level 3 (HSM).
- Kunci eksternal di manajer kunci eksternal Anda.

Enkripsi ganda memiliki efek sebagai berikut:

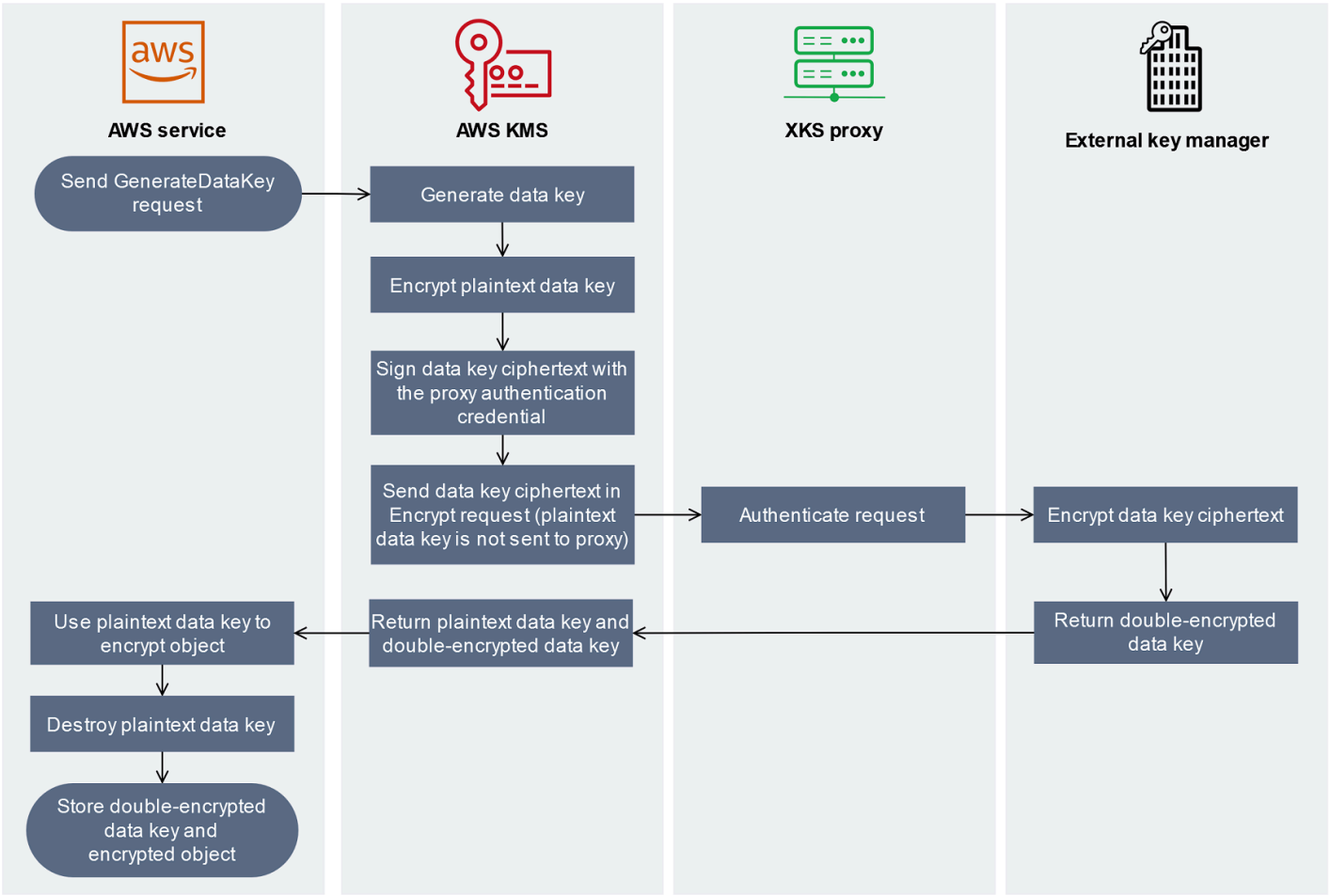
- AWS KMS tidak dapat mendekripsi ciphertext apa pun yang dienkripsi oleh kunci KMS di toko kunci eksternal tanpa akses ke kunci eksternal Anda melalui proxy penyimpanan kunci eksternal Anda.
- Anda tidak dapat mendekripsi ciphertext apa pun yang dienkripsi oleh kunci KMS di penyimpanan kunci eksternal di luar AWS, bahkan jika Anda memiliki materi kunci eksternal.
- Anda tidak dapat membuat ulang kunci KMS yang telah dihapus dari penyimpanan kunci eksternal, bahkan jika Anda memiliki materi kunci eksternal. Setiap kunci KMS memiliki metadata unik yang disertakan dalam ciphertext simetris. Kunci KMS baru tidak akan dapat mendekripsi ciphertext yang dienkripsi oleh kunci asli, bahkan jika menggunakan bahan kunci eksternal yang sama.

Untuk contoh enkripsi ganda dalam praktiknya, lihatCara kerja toko kunci eksternal.

Cara kerja toko kunci eksternal

Penyimpanan kunci eksternal Anda, proxy penyimpanan kunci eksternal, dan manajer kunci eksternal bekerja sama untuk melindungi AWS sumber daya Anda. Prosedur berikut menggambarkan alur kerja enkripsi tipikal Layanan AWS yang mengenkripsi setiap objek di bawah kunci data unik yang dilindungi oleh kunci KMS. Dalam hal ini, Anda telah memilih kunci KMS di toko kunci eksternal untuk melindungi objek. Contoh ini menunjukkan bagaimana AWS KMS menggunakan enkripsi ganda untuk melindungi kunci data dalam perjalanan dan memastikan bahwa ciphertext yang dihasilkan oleh kunci KMS di penyimpanan kunci eksternal selalu setidaknya sekuat ciphertext yang dienkripsi oleh kunci KMS simetris standar dengan materi kunci di dalamnya. AWS KMS

Metode enkripsi yang digunakan oleh masing-masing aktual Layanan AWS yang terintegrasi dengan AWS KMS variatif. Untuk detailnya, lihat topik “Perlindungan data” di bagian Keamanan Layanan AWS dokumentasi.



1. Anda menambahkan objek baru ke Layanan AWS sumber daya Anda. Untuk mengenkripsi objek, Layanan AWS mengirimkan GenerateDataKeypermintaan untuk AWS KMS menggunakan kunci KMS di toko kunci eksternal Anda.
2. AWS KMS menghasilkan kunci data simetris 256-bit dan bersiap untuk mengirim salinan kunci data teks biasa ke manajer kunci eksternal Anda melalui proxy penyimpanan kunci eksternal Anda. AWS KMS memulai proses enkripsi ganda dengan mengenkripsi kunci data plaintext dengan materi kunci yang terkait dengan AWS KMS kunci KMS di penyimpanan kunci eksternal.
3. AWS KMS mengirimkan permintaan enkripsi ke proxy penyimpanan kunci eksternal yang terkait dengan penyimpanan kunci eksternal. Permintaan termasuk ciphertext kunci data yang akan dienkripsi dan ID kunci eksternal yang terkait dengan kunci KMS. AWS KMS menandatangani permintaan menggunakan kredensi otentikasi proxy untuk proxy penyimpanan kunci eksternal Anda.

   Salinan plaintext dari kunci data tidak dikirim ke proxy penyimpanan kunci eksternal.

4. Proxy penyimpanan kunci eksternal mengotentikasi permintaan, dan kemudian meneruskan permintaan enkripsi ke manajer kunci eksternal Anda.

   Beberapa proxy penyimpanan kunci eksternal juga menerapkan kebijakan otorisasi opsional yang memungkinkan hanya prinsipal yang dipilih untuk melakukan operasi dalam kondisi tertentu.

5. Manajer kunci eksternal Anda mengenkripsi ciphertext kunci data menggunakan kunci eksternal yang ditentukan. Manajer kunci eksternal mengembalikan kunci data terenkripsi ganda ke proxy penyimpanan kunci eksternal Anda, yang mengembalikannya ke. AWS KMS

6. AWS KMS mengembalikan kunci data plaintext dan salinan terenkripsi ganda dari kunci data tersebut ke file. Layanan AWS

7. Layanan AWS Menggunakan kunci data plaintext untuk mengenkripsi objek sumber daya, menghancurkan kunci data plaintext, dan menyimpan kunci data terenkripsi dengan objek terenkripsi.

   Beberapa Layanan AWS mungkin cache kunci data plaintext untuk digunakan untuk beberapa objek, atau untuk digunakan kembali saat sumber daya sedang digunakan. Lihat perinciannya di Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Untuk mendekripsi objek terenkripsi, Layanan AWS harus mengirim kunci data terenkripsi kembali ke dalam permintaan Dekripsi. AWS KMS Untuk mendekripsi kunci data terenkripsi, AWS KMS harus mengirim kunci data terenkripsi kembali ke proxy penyimpanan kunci eksternal Anda dengan ID kunci eksternal. Jika permintaan dekripsi ke proxy penyimpanan kunci eksternal gagal karena alasan apa pun, AWS KMS tidak dapat mendekripsi kunci data terenkripsi, dan tidak dapat mendekripsi objek terenkripsi. Layanan AWS

Mengontrol akses ke toko kunci eksternal Anda

Semua fitur kontrol AWS KMS akses — kebijakan utama, kebijakan IAM, dan hibah — yang Anda gunakan dengan kunci KMS standar, bekerja dengan cara yang sama untuk kunci KMS di penyimpanan kunci eksternal. Anda dapat menggunakan kebijakan IAM untuk mengontrol akses ke operasi API yang membuat dan mengelola penyimpanan kunci eksternal. Anda menggunakan kebijakan IAM dan kebijakan utama untuk mengontrol akses ke AWS KMS keys penyimpanan kunci eksternal Anda. Anda juga dapat menggunakan kebijakan kontrol layanan untuk AWS organisasi dan kebijakan titik akhir VPC Anda untuk mengontrol akses ke kunci KMS di penyimpanan kunci eksternal Anda.

Sebaiknya Anda hanya memberikan izin kepada pengguna dan peran yang mereka perlukan untuk tugas yang mungkin mereka lakukan.

Topik

Mengotorisasi manajer toko kunci eksternal

Prinsipal yang membuat dan mengelola penyimpanan kunci eksternal memerlukan izin untuk operasi penyimpanan kunci kustom. Daftar berikut menjelaskan izin minimum yang diperlukan untuk pengelola penyimpanan kunci eksternal. Karena penyimpanan kunci khusus bukan AWS sumber daya, Anda tidak dapat memberikan izin ke penyimpanan kunci eksternal untuk prinsipal di tempat lain. Akun AWS

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms:DeleteCustomKeyStore`

Prinsipal yang membuat penyimpanan kunci eksternal memerlukan izin untuk membuat dan mengkonfigurasi komponen penyimpanan kunci eksternal. Prinsipal dapat membuat toko kunci eksternal hanya di akun mereka sendiri. Untuk membuat penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC, kepala sekolah harus memiliki izin untuk membuat komponen berikut:

- VPC Amazon
- Subnet publik dan pribadi
- Penyeimbang beban jaringan dan kelompok sasaran
- Layanan titik akhir Amazon VPC

Untuk detailnya, lihat Manajemen identitas dan akses untuk VPC Amazon, Identitas, dan manajemen akses untuk titik akhir VPC dan layanan titik akhir VPC serta izin API Elastic Load Balancing.

Mengotorisasi pengguna kunci KMS di toko kunci eksternal

Prinsipal yang membuat dan mengelola AWS KMS keys di toko kunci eksternal Anda memerlukan izin yang sama dengan mereka yang membuat dan mengelola kunci KMS apa pun. AWS KMS Kebijakan kunci default untuk kunci KMS di penyimpanan kunci eksternal identik dengan kebijakan kunci default untuk kunci KMS di. AWS KMS Attribute-based access control (ABAC), yang menggunakan tag dan alias untuk mengontrol akses ke kunci KMS, juga efektif pada kunci KMS di toko kunci eksternal.

Prinsipal yang menggunakan kunci KMS di toko kunci kustom Anda untuk operasi kriptografi memerlukan izin untuk melakukan operasi kriptografi dengan kunci KMS, seperti KMS: Decrypt. Anda dapat memberikan izin ini di IAM atau kebijakan kunci. Namun, mereka tidak memerlukan izin tambahan untuk menggunakan kunci KMS di toko kunci khusus.

Untuk menetapkan izin yang hanya berlaku untuk kunci KMS di penyimpanan kunci eksternal, gunakan kondisi `kms:KeyOrigin`kebijakan dengan nilai. `EXTERNAL_KEY_STORE` Anda dapat menggunakan kondisi ini untuk membatasi izin kms: atau CreateKey izin apa pun yang khusus untuk sumber daya kunci KMS. Misalnya, kebijakan IAM berikut memungkinkan identitas yang dilampirkan untuk memanggil operasi yang ditentukan pada semua kunci KMS di akun, asalkan kunci KMS berada di penyimpanan kunci eksternal. Perhatikan bahwa Anda dapat membatasi izin ke kunci KMS di toko kunci eksternal, dan kunci KMS diAkun AWS, tetapi tidak untuk penyimpanan kunci eksternal tertentu di akun.

```
{
  "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"
    }
```

```
  }
}
```

Otorisasi AWS KMS untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda

AWS KMSberkomunikasi dengan pengelola kunci eksternal Anda hanya melalui proxy penyimpanan kunci eksternal yang Anda berikan. AWS KMSmengautentikasi proxy Anda dengan menandatangani permintaannya menggunakan proses Signature Version 4 (SigV4) dengan kredensi otentikasi proxy penyimpanan kunci eksternal yang Anda tentukan. Jika Anda menggunakan konektivitas titik akhir publik untuk proxy penyimpanan kunci eksternal Anda, AWS KMS tidak memerlukan izin tambahan.

Namun, jika Anda menggunakan konektivitas layanan titik akhir VPC, Anda harus memberikan AWS KMS izin untuk membuat titik akhir antarmuka ke layanan endpoint Amazon VPC Anda. Izin ini diperlukan terlepas dari apakah proxy penyimpanan kunci eksternal ada di VPC Anda atau proxy penyimpanan kunci eksternal berada di tempat lain, tetapi menggunakan layanan titik akhir VPC untuk berkomunikasi dengannya. AWS KMS

AWS KMSUntuk memungkinkan membuat titik akhir antarmuka, gunakan konsol VPC Amazon atau ModifyVpcEndpointServicePermissionsoperasinya. Izinkan izin untuk prinsipal berikut:`cks.kms.<region>.amazonaws.com`.

Misalnya, AWS CLI perintah berikut memungkinkan AWS KMS untuk terhubung ke layanan titik akhir VPC yang ditentukan di Wilayah AS Barat (Oregon) (us-barat-2). Sebelum menggunakan perintah ini, ganti ID layanan Amazon VPC dan Wilayah AWS dengan nilai yang valid untuk konfigurasi Anda.

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Untuk menghapus izin ini, gunakan konsol VPC Amazon atau ModifyVpcEndpointServicePermissionsdengan parameter. `RemoveAllowedPrincipals`

Otorisasi proxy penyimpanan kunci eksternal (opsional)

Beberapa proxy penyimpanan kunci eksternal menerapkan persyaratan otorisasi untuk penggunaan kunci eksternalnya. Proxy penyimpanan kunci eksternal diizinkan, tetapi tidak diperlukan, untuk merancang dan menerapkan skema otorisasi yang memungkinkan pengguna tertentu untuk meminta operasi tertentu hanya dalam kondisi tertentu. Misalnya, proxy mungkin dikonfigurasi untuk memungkinkan pengguna A mengenkripsi dengan kunci eksternal tertentu, tetapi tidak untuk mendekripsi dengannya.

Otorisasi proxy independen dari otentikasi proxy berbasis SIGV4 yang AWS KMS memerlukan semua proxy penyimpanan kunci eksternal. Ini juga independen dari kebijakan utama, kebijakan IAM, dan hibah yang mengotorisasi akses ke operasi yang mempengaruhi penyimpanan kunci eksternal atau kunci KMS-nya.

Untuk mengaktifkan otorisasi oleh proxy penyimpanan kunci eksternal, AWS KMS sertakan metadata di setiap permintaan API proxy, termasuk pemanggil, kunci KMS, AWS KMS operasi, (jika ada). Layanan AWS Metadata permintaan untuk versi 1 (v1) dari API proxy kunci eksternal adalah sebagai berikut.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Misalnya, Anda dapat mengonfigurasi proxy Anda untuk mengizinkan permintaan dari prinsipal tertentu (`awsPrincipalArn`), tetapi hanya jika permintaan dibuat atas nama prinsipal oleh Layanan AWS (`kmsViaService`) tertentu.

Jika otorisasi proxy gagal, AWS KMS operasi terkait gagal dengan pesan yang menjelaskan kesalahan. Untuk detailnya, lihat Masalah otorisasi proxy.

otentikasi mTLS (opsional)

Untuk mengaktifkan proxy penyimpanan kunci eksternal Anda untuk mengautentikasi permintaanAWS KMS, AWS KMS tandatangani semua permintaan ke proxy penyimpanan kunci eksternal Anda dengan kredensi otentikasi proxy Signature V4 (SigV4) untuk penyimpanan kunci eksternal Anda.

Untuk memberikan jaminan tambahan bahwa proxy penyimpanan kunci eksternal Anda hanya merespons AWS KMS permintaan, beberapa proxy kunci eksternal mendukung mutual Transport Layer Security (mTLS), di mana kedua pihak dalam transaksi menggunakan sertifikat untuk mengautentikasi satu sama lain. mTLS menambahkan otentikasi sisi klien — di mana server proxy penyimpanan kunci eksternal mengautentikasi klien — ke otentikasi sisi server yang disediakan TLS standar. AWS KMS Dalam kasus yang jarang terjadi bahwa kredensi otentikasi proxy Anda

dikompromikan, mTL mencegah pihak ketiga membuat permintaan API yang berhasil ke proxy penyimpanan kunci eksternal.

Untuk mengimplementasikan mTL, konfigurasikan proxy penyimpanan kunci eksternal Anda untuk hanya menerima sertifikat TLS sisi klien dengan properti berikut:

- Nama umum subjek pada sertifikat TLS haruscks.kms.*<Region>*.amazonaws.com, misalnya,cks.kms.*eu-west-3*.amazonaws.com.
- Sertifikat harus dirantai ke otoritas sertifikat yang terkait dengan Amazon Trust Services.

Merencanakan toko kunci eksternal

Sebelum membuat penyimpanan kunci eksternal Anda, pilih opsi konektivitas yang menentukan cara AWS KMS berkomunikasi dengan komponen penyimpanan kunci eksternal Anda. Opsi konektivitas yang Anda pilih menentukan sisa proses perencanaan.

Pelajari lebih lanjut:

- Tinjau proses untuk membuat toko kunci eksternal, termasuk merakit prasyarat. Ini akan membantu Anda memastikan bahwa Anda memiliki semua komponen yang Anda butuhkan saat membuat toko kunci eksternal Anda.
- Pelajari cara mengontrol akses ke penyimpanan kunci eksternal Anda, termasuk izin yang diperlukan oleh administrator dan pengguna penyimpanan kunci eksternal.
- Pelajari tentang CloudWatch metrik dan dimensi Amazon yang AWS KMS direkam untuk penyimpanan kunci eksternal. Kami sangat menyarankan agar Anda membuat alarm untuk memantau penyimpanan kunci eksternal Anda sehingga Anda dapat mendeteksi tanda-tanda awal masalah kinerja dan operasional.

Memilih opsi konektivitas proxy

Jika Anda membuat penyimpanan kunci eksternal, Anda perlu menentukan bagaimana AWS KMS berkomunikasi dengan proxy penyimpanan kunci eksternal Anda. Pilihan ini akan menentukan komponen mana yang Anda butuhkan dan bagaimana Anda mengkonfigurasinya. AWS KMSmendukung opsi konektivitas berikut. Pilih opsi yang memenuhi tujuan kinerja dan keamanan Anda.

Sebelum Anda mulai, konfirmasikan bahwa Anda memerlukan toko kunci eksternal. Sebagian besar pelanggan dapat menggunakan kunci KMS yang didukung oleh materi AWS KMS utama.

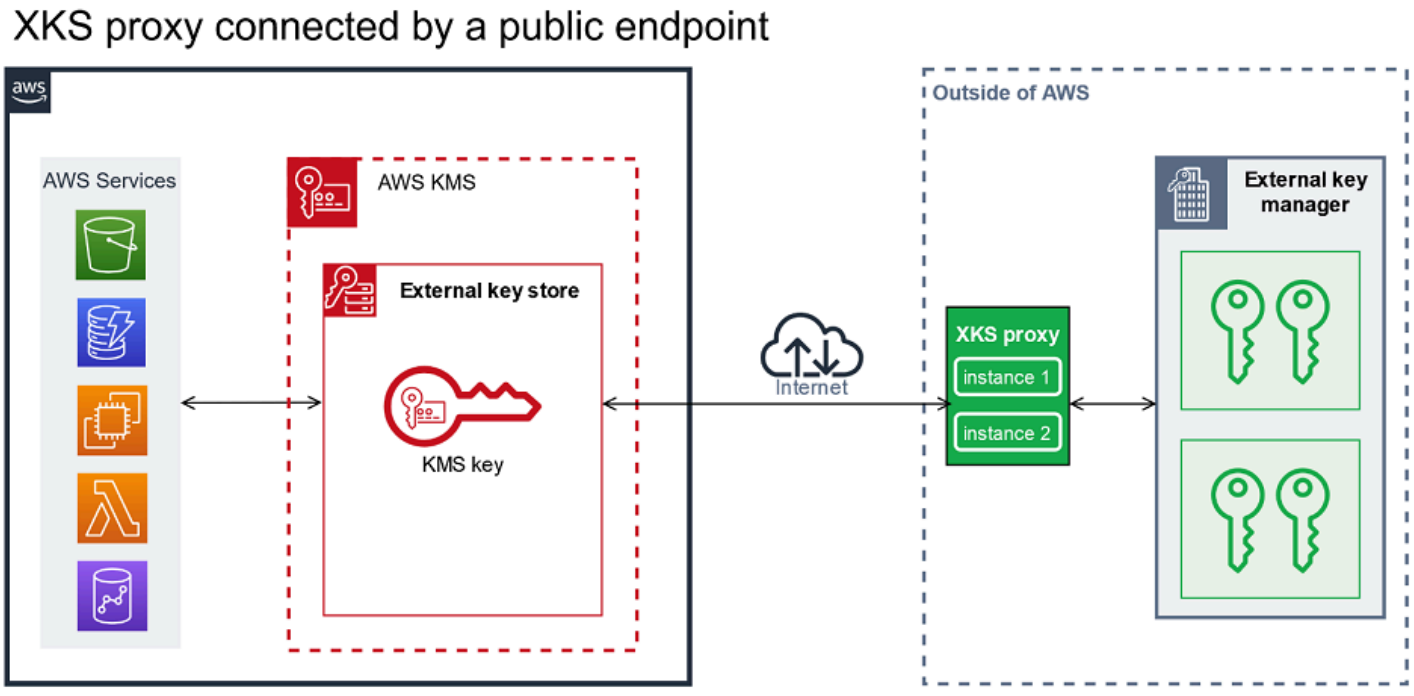
> **Note**
>
> Jika proxy penyimpanan kunci eksternal Anda dibangun ke dalam pengelola kunci eksternal Anda, konektivitas Anda mungkin telah ditentukan sebelumnya. Untuk panduan, lihat dokumentasi untuk pengelola kunci eksternal atau proxy penyimpanan kunci eksternal Anda.

Anda dapat mengubah opsi konektivitas proxy penyimpanan kunci eksternal Anda bahkan di toko kunci eksternal yang beroperasi. Namun, prosesnya harus direncanakan dan dijalankan dengan hati-hati untuk meminimalkan gangguan, menghindari kesalahan, dan memastikan akses berkelanjutan ke kunci kriptografi yang mengenkripsi data Anda.

Konektivitas titik akhir publik

AWS KMSterhubung ke proxy penyimpanan kunci eksternal (proxy XKS) melalui internet menggunakan titik akhir publik.

Opsi konektivitas ini lebih mudah diatur dan dipelihara, dan selaras dengan beberapa model manajemen kunci. Namun, itu mungkin tidak memenuhi persyaratan keamanan beberapa organisasi.



Persyaratan

Jika Anda memilih konektivitas titik akhir publik, berikut ini diperlukan.

- Proxy penyimpanan kunci eksternal Anda harus dapat dijangkau pada titik akhir yang dapat dirutekan secara publik.
- Anda dapat menggunakan titik akhir publik yang sama untuk beberapa penyimpanan kunci eksternal asalkan mereka menggunakan nilai jalur URI proxy yang berbeda.
- Anda tidak dapat menggunakan titik akhir yang sama untuk penyimpanan kunci eksternal dengan konektivitas titik akhir publik dan penyimpanan kunci eksternal apa pun dengan konektivitas layanan titik akhir VPC yang samaWilayah AWS, meskipun penyimpanan kunci berbeda. Akun AWS
- Anda harus mendapatkan sertifikat TLS yang dikeluarkan oleh otoritas sertifikat publik yang didukung untuk toko kunci eksternal. Untuk daftar, lihat Otoritas Sertifikat Tepercaya.

  Nama umum subjek (CN) pada sertifikat TLS harus cocok dengan nama domain di titik akhir URI proxy untuk proxy penyimpanan kunci eksternal. Misalnya, jika titik akhir publik adalah`https://myproxy.xks.example.com`, TLS, CN pada sertifikat TLS harus atau. `myproxy.xks.example.com` `*.xks.example.com`
- Pastikan bahwa setiap firewall antara AWS KMS dan proxy penyimpanan kunci eksternal memungkinkan lalu lintas ke dan dari port 443 pada proxy. AWS KMSberkomunikasi di port 443. Nilai ini tidak dapat dikonfigurasi.

Untuk semua persyaratan untuk penyimpanan kunci eksternal, lihat Merakit prasyarat.

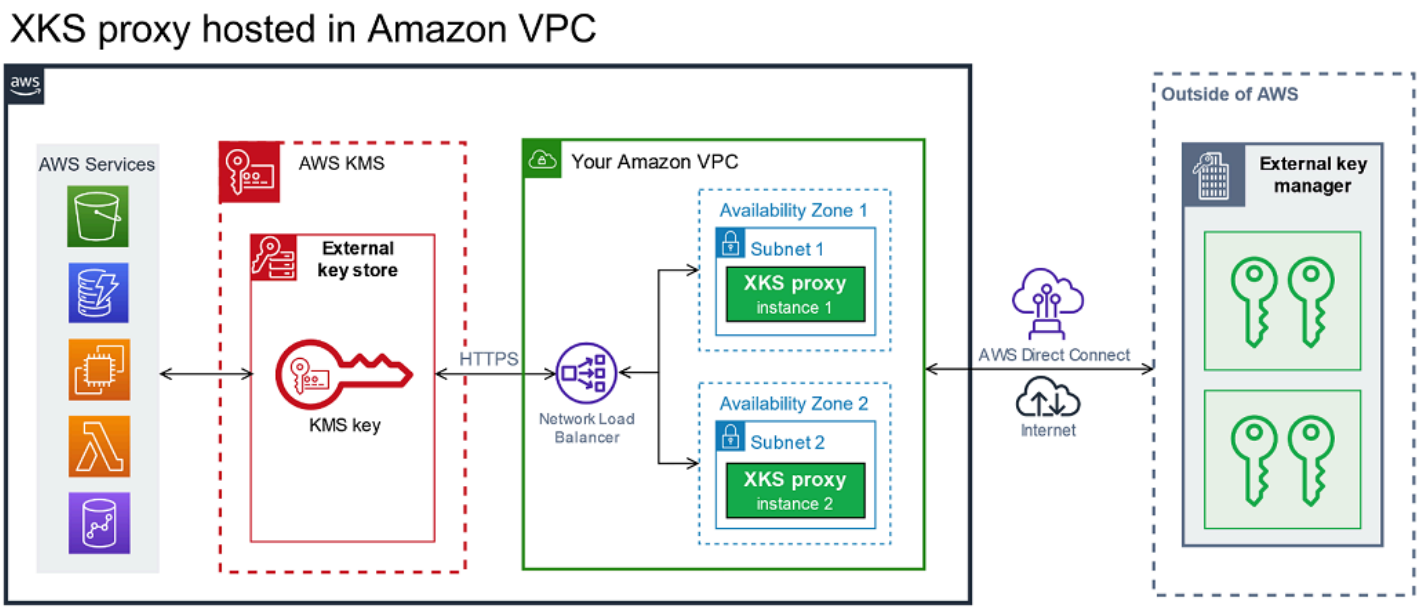
Konektivitas layanan titik akhir VPC

AWS KMSterhubung ke proxy penyimpanan kunci eksternal (proxy XKS) dengan membuat titik akhir antarmuka ke layanan titik akhir VPC Amazon yang Anda buat dan konfigurasikan. Anda bertanggung jawab untuk membuat layanan titik akhir VPC dan menghubungkan VPC Anda ke pengelola kunci eksternal Anda.

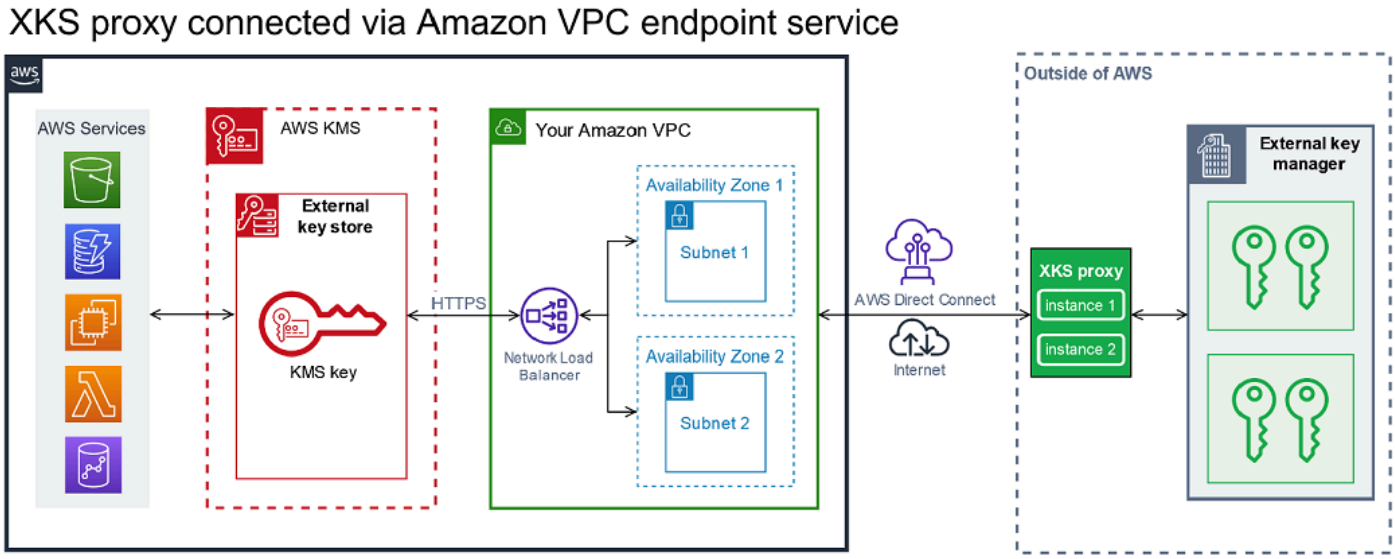
Layanan endpoint Anda dapat menggunakan salah satu opsi VPC Jaringan-ke-Amazon yang didukung untuk komunikasi, termasuk. AWS Direct Connect

Opsi konektivitas ini lebih rumit untuk diatur dan dipelihara. Tetapi menggunakanAWS PrivateLink, yang memungkinkan AWS KMS untuk terhubung secara pribadi ke VPC Amazon Anda dan proxy toko kunci eksternal Anda tanpa menggunakan internet publik.

Anda dapat menemukan proxy toko kunci eksternal Anda di VPC Amazon Anda.



Atau, cari proxy penyimpanan kunci eksternal Anda di luar AWS dan gunakan layanan titik akhir VPC Amazon Anda hanya untuk komunikasi yang aman. AWS KMS



Mengkonfigurasi konektivitas layanan titik akhir VPC

Gunakan panduan di bagian ini untuk membuat dan mengonfigurasi AWS sumber daya dan komponen terkait yang diperlukan untuk penyimpanan kunci eksternal yang menggunakan konektivitas layanan titik akhir VPC. Sumber daya yang terdaftar untuk opsi konektivitas ini adalah suplemen untuk sumber daya yang diperlukan untuk semua toko kunci eksternal. Setelah Anda membuat dan mengkonfigurasi sumber daya yang diperlukan, Anda dapat membuat penyimpanan kunci eksternal Anda.

Anda dapat menemukan proxy penyimpanan kunci eksternal di VPC Amazon atau menemukan proxy di luar AWS dan menggunakan layanan titik akhir VPC Anda untuk komunikasi.

Sebelum Anda mulai, konfirmasikan bahwa Anda memerlukan toko kunci eksternal. Sebagian besar pelanggan dapat menggunakan kunci KMS yang didukung oleh materi AWS KMS utama.

> **Note**
>
> Beberapa elemen yang diperlukan untuk konektivitas layanan titik akhir VPC mungkin disertakan dalam pengelola kunci eksternal Anda. Selain itu, perangkat lunak Anda mungkin memiliki persyaratan konfigurasi tambahan. Sebelum membuat dan mengonfigurasi AWS sumber daya di bagian ini, lihat dokumentasi proxy dan manajer kunci Anda.

Topik

Persyaratan untuk konektivitas layanan titik akhir VPC

Jika Anda memilih konektivitas layanan titik akhir VPC untuk penyimpanan kunci eksternal Anda, sumber daya berikut diperlukan.

Untuk meminimalkan latensi jaringan, buat AWS komponen Anda di bagian yang didukung Wilayah AWS yang paling dekat dengan pengelola kunci eksternal Anda. Jika memungkinkan, pilih Wilayah dengan waktu pulang-pergi jaringan (RTT) 35 milidetik atau kurang.

- VPC Amazon yang terhubung ke pengelola kunci eksternal Anda. Itu harus memiliki setidaknya dua subnet pribadi di dua Availability Zone yang berbeda.

  Anda dapat menggunakan VPC Amazon yang ada untuk toko kunci eksternal Anda, asalkan memenuhi persyaratan untuk digunakan dengan toko kunci eksternal. Beberapa toko kunci

eksternal dapat berbagi VPC Amazon, tetapi setiap toko kunci eksternal harus memiliki layanan titik akhir VPC sendiri dan nama DNS pribadi.

- Layanan endpoint VPC Amazon yang didukung oleh AWS PrivateLink penyeimbang beban jaringan dan grup target.

  Layanan endpoint tidak dapat memerlukan penerimaan. Juga, Anda harus menambahkan AWS KMS sebagai kepala sekolah yang diizinkan. Ini memungkinkan AWS KMS untuk membuat titik akhir antarmuka sehingga dapat berkomunikasi dengan proxy penyimpanan kunci eksternal Anda.
- Nama DNS pribadi untuk layanan titik akhir VPC yang unik di dalamnya. Wilayah AWS

  Nama DNS pribadi harus merupakan subdomain dari domain publik tingkat yang lebih tinggi. Misalnya, jika nama DNS pribadi adalah`myproxy-private.xks.example.com`, itu harus menjadi subdomain dari domain publik seperti `xks.example.com` atau. `example.com`

  Anda harus memverifikasi kepemilikan domain DNS untuk nama DNS pribadi.
- Sertifikat TLS yang dikeluarkan oleh otoritas sertifikat publik yang didukung untuk proxy penyimpanan kunci eksternal Anda.

  Nama umum subjek (CN) pada sertifikat TLS harus cocok dengan nama DNS pribadi. Misalnya, jika nama DNS pribadi adalah`myproxy-private.xks.example.com`, CN pada sertifikat TLS harus atau. `myproxy-private.xks.example.com` `*.xks.example.com`

Untuk semua persyaratan untuk penyimpanan kunci eksternal, lihat Merakit prasyarat.

Membuat Amazon VPC dan subnet

Konektivitas layanan titik akhir VPC memerlukan VPC Amazon yang terhubung ke pengelola kunci eksternal Anda dengan setidaknya dua subnet pribadi. Anda dapat membuat VPC Amazon atau menggunakan VPC Amazon yang sudah ada yang memenuhi persyaratan untuk toko kunci eksternal. Untuk bantuan dalam membuat VPC Amazon baru, lihat Membuat VPC di Panduan Pengguna Amazon Virtual Private Cloud.

Persyaratan untuk VPC Amazon Anda

Untuk bekerja dengan toko kunci eksternal menggunakan konektivitas layanan titik akhir VPC, VPC Amazon harus memiliki properti berikut:

- Harus berada di Wilayah yang sama Akun AWS dan didukung sebagai toko kunci eksternal Anda.
- Memerlukan setidaknya dua subnet pribadi, masing-masing di Availability Zone yang berbeda.

- Rentang alamat IP pribadi VPC Amazon Anda tidak boleh tumpang tindih dengan rentang alamat IP pribadi dari pusat data yang menghosting manajer kunci eksternal Anda.
- Semua komponen harus menggunakan IPv4.

Anda memiliki banyak opsi untuk menghubungkan VPC Amazon ke proxy penyimpanan kunci eksternal Anda. Pilih opsi yang memenuhi kebutuhan kinerja dan keamanan Anda. Untuk daftar, lihat Menghubungkan VPC Anda ke jaringan lain dan opsi konektivitas VPC Jaringan-ke-Amazon. Untuk detail selengkapnya AWS Direct Connect, lihat, dan Panduan AWS Site-to-Site VPN Pengguna.

Membuat VPC Amazon untuk toko kunci eksternal Anda

Gunakan petunjuk berikut untuk membuat VPC Amazon untuk toko kunci eksternal Anda. VPC Amazon hanya diperlukan jika Anda memilih opsi konektivitas layanan titik akhir VPC. Anda dapat menggunakan VPC Amazon yang sudah ada yang memenuhi persyaratan untuk penyimpanan kunci eksternal.

Ikuti petunjuk dalam topik Buat VPC, subnet, dan sumber daya VPC lainnya menggunakan nilai wajib berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Blok CIDR IPv4	Masukkan alamat IP untuk VPC Anda. Rentang alamat IP pribadi VPC Amazon Anda tidak boleh tumpang tindih dengan rentang alamat IP pribadi dari pusat data yang menghosting manajer kunci eksternal Anda.
Jumlah Availability Zone (AZ)	2 atau lebih
Jumlah subnet publik	Tidak ada yang diperlukan (0)
Jumlah subnet pribadi	Satu untuk setiap AZ
Gateway NAT	Tidak ada yang diperlukan.
Titik akhir VPC	Tidak ada yang diperlukan.

Bidang	Nilai
Aktifkan nama host DNS	Ya
Aktifkan resolusi DNS	Ya

Pastikan untuk menguji komunikasi VPC Anda. Misalnya, jika proxy penyimpanan kunci eksternal Anda tidak terletak di VPC Amazon Anda, buat instans Amazon EC2 di VPC Amazon Anda, verifikasi bahwa VPC Amazon dapat berkomunikasi dengan proxy penyimpanan kunci eksternal Anda.

Menghubungkan VPC ke manajer kunci eksternal

Hubungkan VPC ke pusat data yang menampung pengelola kunci eksternal Anda menggunakan salah satu opsi konektivitas jaringan yang didukung Amazon VPC. Pastikan instans Amazon EC2 di VPC (atau proxy penyimpanan kunci eksternal, jika ada di VPC), dapat berkomunikasi dengan pusat data dan pengelola kunci eksternal.

Membuat grup target

Sebelum Anda membuat layanan endpoint VPC yang diperlukan, buat komponen yang diperlukan, network load balancer (NLB) dan grup target. Network Load Balancer (NLB) mendistribusikan permintaan di antara beberapa target sehat, yang mana pun dapat melayani permintaan. Pada langkah ini, Anda membuat grup target dengan setidaknya dua host untuk proxy penyimpanan kunci eksternal Anda, dan mendaftarkan alamat IP Anda dengan grup target.

Ikuti petunjuk dalam Mengkonfigurasi topik grup target menggunakan nilai wajib berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Tipe target	Alamat IP
Protokol	TCP
Port	443
Jenis alamat IP	IPv4

Bidang	Nilai
VPC	Pilih VPC tempat Anda akan membuat layanan titik akhir VPC untuk toko kunci eksternal Anda.
Protokol dan jalur pemeriksaan kesehatan	Protokol dan jalur pemeriksaan kesehatan Anda akan berbeda dengan konfigurasi proxy penyimpanan kunci eksternal Anda. Konsultasikan dokumentasi untuk pengelola kunci eksternal atau proxy penyimpanan kunci eksternal Anda. Untuk informasi umum tentang mengonfigurasi pemeriksaan kesehatan untuk grup target Anda, lihat Pemeriksaan Kesehatan untuk grup target Anda di Panduan Pengguna Elastic Load Balancing untuk Network Load Balancers.
Jaringan	Alamat IP pribadi lainnya
Alamat IPv4	Alamat pribadi proxy toko kunci eksternal Anda
Port	443

Membuat penyeimbang beban jaringan

Penyeimbang beban jaringan mendistribusikan lalu lintas jaringan, termasuk permintaan dari AWS KMS proxy penyimpanan kunci eksternal Anda, ke target yang dikonfigurasi.

Ikuti petunjuk dalam Konfigurasi penyeimbang beban dan topik pendengar untuk mengonfigurasi dan menambahkan pendengar dan membuat penyeimbang beban menggunakan nilai wajib berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Skema	Internal
Jenis alamat IP	IPv4
Pemetaan jaringan	Pilih VPC tempat Anda akan membuat layanan titik akhir VPC untuk toko kunci eksternal Anda.
Pemetaan	Pilih kedua zona ketersediaan (setidaknya dua) yang Anda konfigurasikan untuk subnet VPC Anda. Verifikasi nama subnet dan alamat IP pribadi.

Bidang	Nilai
Protokol	TCP
Port	443
Tindakan default: Teruskan ke	Pilih grup target untuk penyeimbang beban jaringan Anda.

Membuat layanan titik akhir VPC

Biasanya, Anda membuat titik akhir ke layanan. Namun, ketika Anda membuat layanan titik akhir VPC, Anda adalah penyedia, dan AWS KMS membuat titik akhir ke layanan Anda. Untuk penyimpanan kunci eksternal, buat layanan titik akhir VPC dengan penyeimbang beban jaringan yang Anda buat di langkah sebelumnya. Layanan titik akhir VPC harus berada di Wilayah yang sama Akun AWS dan didukung sebagai penyimpanan kunci eksternal Anda.

Beberapa toko kunci eksternal dapat berbagi VPC Amazon, tetapi setiap toko kunci eksternal harus memiliki layanan titik akhir VPC sendiri dan nama DNS pribadi.

Ikuti petunjuk dalam topik Create an endpoint service untuk membuat layanan endpoint VPC Anda dengan nilai wajib berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Tipe penyeimbang beban	Jaringan
Penyeimbang beban yang tersedia	Pilih penyeimbang beban jaringan yang Anda buat pada langkah sebelumnya. Jika penyeimbang beban baru Anda tidak muncul dalam daftar, verifikasi bahwa statusnya aktif. Mungkin perlu beberapa menit agar status penyeimbang beban berubah dari penyediaan menjadi aktif.
Penerimaan diperlukan	Salah. Hapus centang pada kotak centang.

Bidang	Nilai
	Tidak memerlukan penerimaan. AWS KMStidak dapat terhubung ke layanan titik akhir VPC tanpa penerimaan manual. Jika penerimaan diperlukan, upaya untuk membuat penyimpanan kunci eksternal gagal dengan `XksProxyInvalidConfigurationException` pengecualian.
Aktifkan nama DNS pribadi	Kaitkan nama DNS pribadi dengan layanan
Nama DNS pribadi	Masukkan nama DNS pribadi yang unik di dalamnyaWilayah AWS. Nama DNS pribadi harus menjadi subdomain dari domain publik tingkat yang lebih tinggi. Misalnya, jika nama DNS pribadi adalah`myproxy-private.xks.example.com`, itu harus menjadi subdomain dari domain publik seperti `xks.example.com` atau. `example.com` Nama DNS pribadi ini harus cocok dengan nama umum subjek (CN) dalam sertifikat TLS yang dikonfigurasi pada proxy penyimpanan kunci eksternal Anda. Misalnya, jika nama DNS pribadi adalah`myproxy-private.xks.example.com`, CN pada sertifikat TLS harus atau. `myproxy-private.xks.example.com` `*.xks.example.com` Jika sertifikat dan nama DNS pribadi tidak cocok, upaya untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal gagal dengan kode kesalahan koneksi. `XKS_PROXY_INVALID_TLS_CONFIGURATION` Untuk detailnya, lihat Kesalahan konfigurasi umum.
Jenis alamat IP yang didukung	IPv4

Memverifikasi domain nama DNS pribadi Anda

Saat Anda membuat layanan titik akhir VPC, status verifikasi domainnya adalah. `pendingVerification` Sebelum menggunakan layanan titik akhir VPC untuk membuat penyimpanan kunci eksternal, status ini harus. `verified` Untuk memverifikasi bahwa Anda memiliki domain yang terkait dengan nama DNS pribadi Anda, Anda harus membuat catatan TXT di server DNS publik.

Misalnya, jika nama DNS pribadi untuk layanan `myproxy-private.xks.example.com` titik akhir VPC Anda, Anda harus membuat catatan TXT di domain publik, `xks.example.com` seperti `example.com` atau, mana pun yang bersifat publik. AWS PrivateLinkmencari catatan TXT terlebih dahulu `xks.example.com` dan kemudian. `example.com`

> **Tip**
>
> Setelah Anda menambahkan catatan TXT, mungkin perlu beberapa menit untuk mengubah nilai status verifikasi Domain dari `pendingVerification` ke. `verify`

Untuk memulai, cari status verifikasi domain Anda menggunakan salah satu metode berikut. Nilai yang valid adalah `verified`, `pendingVerification`, dan `failed`.

- Di konsol VPC Amazon, pilih layanan Endpoint, dan pilih layanan endpoint Anda. Di panel detail, lihat Status verifikasi domain.
- Gunakan DescribeVpcEndpointServiceConfigurationsoperasi. `State`Nilainya ada di `ServiceConfigurations.PrivateDnsNameConfiguration.State` lapangan.

Jika status verifikasi tidak`verified`, ikuti petunjuk dalam topik verifikasi kepemilikan Domain untuk menambahkan catatan TXT ke server DNS domain Anda dan verifikasi bahwa catatan TXT diterbitkan. Kemudian periksa kembali status verifikasi Anda.

Anda tidak diharuskan membuat catatan A untuk nama domain DNS pribadi. Saat AWS KMS membuat titik akhir antarmuka ke layanan titik akhir VPC AndaAWS PrivateLink, secara otomatis membuat zona yang dihosting dengan catatan A yang diperlukan untuk nama domain pribadi di VPC. AWS KMS Untuk penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC, ini terjadi ketika Anda menghubungkan penyimpanan kunci eksternal Anda ke proxy penyimpanan kunci eksternal.

Otorisasi AWS KMS untuk terhubung ke layanan titik akhir VPC

Anda harus menambahkan AWS KMS ke daftar Izinkan prinsipal untuk layanan titik akhir VPC Anda. Ini memungkinkan AWS KMS untuk membuat titik akhir antarmuka ke layanan titik akhir VPC Anda. Jika AWS KMS bukan prinsipal yang diizinkan, upaya untuk membuat penyimpanan kunci eksternal akan gagal dengan `XksProxyVpcEndpointServiceNotFoundException` pengecualian.

Ikuti petunjuk dalam topik Kelola izin di AWS PrivateLinkPanduan. Gunakan nilai yang diperlukan berikut.

Bidang	Nilai
ARN	`cks.kms.<region>.amazonaws.com` Misalnya, `cks.kms.us-east-1.amazonaws.com`

Selanjutnya: Membuat toko kunci eksternal

Mengelola toko kunci eksternal

Anda dapat mengelola penyimpanan kunci eksternal dengan menggunakan AWS KMS konsol atau AWS KMS API. Anda dapat membuat penyimpanan kunci eksternal, melihat dan mengedit propertinya, memantau kinerjanya, dan menghubungkan dan memutusnya dari proxy penyimpanan kunci eksternal, dan menghapus toko kunci eksternal.

Topik

- Membuat toko kunci eksternal
- Mengedit properti penyimpanan kunci eksternal
- Melihat toko kunci eksternal
- Memantau toko kunci eksternal
- Menghubungkan dan memutuskan penyimpanan kunci eksternal
- Menghapus penyimpanan kunci eksternal

Membuat toko kunci eksternal

Anda dapat membuat satu atau banyak toko kunci eksternal di masing-masing Akun AWS dan Wilayah. Setiap penyimpanan kunci eksternal harus dikaitkan dengan pengelola kunci eksternal di luarAWS, dan proxy penyimpanan kunci eksternal (proxy XKS) yang memediasi komunikasi antara AWS KMS dan manajer kunci eksternal Anda. Untuk detailnya, lihat Merencanakan toko kunci eksternal. Sebelum Anda mulai, konfirmasikan bahwa Anda memerlukan toko kunci eksternal. Sebagian besar pelanggan dapat menggunakan kunci KMS yang didukung oleh materi AWS KMS utama.

> **Tip**
>
> Beberapa manajer kunci eksternal menyediakan metode yang lebih sederhana untuk membuat penyimpanan kunci eksternal. Untuk detailnya, lihat dokumentasi pengelola kunci eksternal Anda.

Sebelum Anda membuat toko kunci eksternal Anda, Anda perlu merakit prasyarat. Selama proses pembuatan, Anda menentukan properti penyimpanan kunci eksternal Anda. Yang terpenting, Anda menunjukkan apakah penyimpanan kunci eksternal Anda AWS KMS menggunakan titik akhir publik atau layanan titik akhir VPC untuk terhubung ke proxy penyimpanan kunci eksternalnya. Anda juga menentukan detail koneksi, termasuk titik akhir URI proxy dan jalur dalam titik akhir proxy tersebut tempat AWS KMS mengirimkan permintaan API ke proxy.

- Jika Anda menggunakan konektivitas titik akhir publik, pastikan itu AWS KMS dapat berkomunikasi dengan proxy Anda melalui internet menggunakan koneksi HTTPS. Ini termasuk mengkonfigurasi TLS pada proxy penyimpanan kunci eksternal dan memastikan bahwa setiap firewall antara AWS KMS dan proxy memungkinkan lalu lintas ke dan dari port 443 pada proxy. Saat membuat penyimpanan kunci eksternal dengan konektivitas titik akhir publik, AWS KMS menguji koneksi dengan mengirimkan permintaan status ke proxy penyimpanan kunci eksternal. Tes ini memverifikasi bahwa titik akhir dapat dijangkau dan proxy penyimpanan kunci eksternal Anda akan menerima permintaan yang ditandatangani dengan kredensi otentikasi proxy penyimpanan kunci eksternal Anda. Jika permintaan pengujian ini gagal, operasi untuk membuat penyimpanan kunci eksternal gagal.
- Jika Anda menggunakan konektivitas layanan titik akhir VPC, pastikan penyeimbang beban jaringan, nama DNS pribadi, dan layanan titik akhir VPC dikonfigurasi dengan benar dan operasional. Jika proxy penyimpanan kunci eksternal tidak ada di VPC, Anda perlu memastikan bahwa layanan titik akhir VPC dapat berkomunikasi dengan proxy penyimpanan kunci eksternal. (AWS KMSmenguji konektivitas layanan titik akhir VPC saat Anda menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal.)

Pertimbangan tambahan:

- AWS KMSmerekam CloudWatch metrik dan dimensi Amazon terutama untuk toko kunci eksternal. Grafik pemantauan berdasarkan beberapa metrik ini muncul di AWS KMS konsol untuk setiap penyimpanan kunci eksternal. Kami sangat menyarankan Anda menggunakan metrik ini untuk membuat alarm yang memantau penyimpanan kunci rexternal Anda. Alarm ini mengingatkan Anda

tentang tanda-tanda awal masalah kinerja dan operasional sebelum terjadi. Untuk petunjuk, lihat Memantau toko kunci eksternal.

- Toko kunci eksternal tunduk pada kuota sumber daya. Penggunaan kunci KMS di toko kunci eksternal tunduk pada kuota permintaan. Tinjau kuota ini sebelum merancang implementasi penyimpanan kunci eksternal Anda.

> **Note**
>
> Tinjau konfigurasi Anda untuk dependensi melingkar yang mungkin mencegahnya berfungsi. Misalnya, jika Anda membuat proxy penyimpanan kunci eksternal menggunakan AWS sumber daya, pastikan bahwa mengoperasikan proxy tidak memerlukan ketersediaan kunci KMS di penyimpanan kunci eksternal yang diakses melalui proxy tersebut.

Semua toko kunci eksternal baru dibuat dalam keadaan terputus. Sebelum Anda dapat membuat kunci KMS toko kunci eksternal Anda, Anda harus menghubungkannya ke proxy penyimpanan kunci eksternal. Untuk mengubah properti penyimpanan kunci eksternal Anda, edit pengaturan penyimpanan kunci eksternal Anda.

Topik

- Memasang prasyarat
- File konfigurasi proxy
- Buat toko kunci eksternal (konsol)
- Buat toko kunci eksternal (API)

Memasang prasyarat

Sebelum Anda membuat penyimpanan kunci eksternal, Anda perlu merakit komponen yang diperlukan, termasuk pengelola kunci eksternal yang akan Anda gunakan untuk mendukung penyimpanan kunci eksternal dan proxy penyimpanan kunci eksternal yang menerjemahkan AWS KMS permintaan ke dalam format yang dapat dipahami oleh manajer kunci eksternal Anda.

Komponen berikut diperlukan untuk semua toko kunci eksternal. Selain komponen ini, Anda perlu menyediakan komponen untuk mendukung opsi konektivitas proxy penyimpanan kunci eksternal yang Anda pilih.

> **Tip**
>
> Manajer kunci eksternal Anda mungkin menyertakan beberapa komponen ini, atau mereka mungkin dikonfigurasi untuk Anda. Untuk detailnya, lihat dokumentasi pengelola kunci eksternal Anda.
> Jika Anda membuat penyimpanan kunci eksternal di AWS KMS konsol, Anda memiliki opsi untuk mengunggah file konfigurasi proxy berbasis JSON yang menentukan jalur URI proxydan kredensi otentikasi proxy. Beberapa proxy penyimpanan kunci eksternal menghasilkan file ini untuk Anda. Untuk detailnya, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau pengelola kunci eksternal.

Manajer kunci eksternal

Setiap penyimpanan kunci eksternal memerlukan setidaknya satu instance pengelola kunci eksternal. Ini bisa berupa modul keamanan perangkat keras fisik atau virtual (HSM), atau perangkat lunak manajemen kunci.

Anda dapat menggunakan satu manajer kunci, tetapi kami merekomendasikan setidaknya dua contoh manajer kunci terkait yang berbagi kunci kriptografi untuk redundansi. Toko kunci eksternal tidak memerlukan penggunaan eksklusif manajer kunci eksternal. Namun, manajer kunci eksternal harus memiliki kapasitas untuk menangani frekuensi yang diharapkan dari enkripsi dan permintaan dekripsi dari AWS layanan yang menggunakan kunci KMS di penyimpanan kunci eksternal untuk melindungi sumber daya Anda. Manajer kunci eksternal Anda harus dikonfigurasi untuk menangani hingga 1800 permintaan per detik dan merespons dalam batas waktu 250 milidetik untuk setiap permintaan. Kami menyarankan Anda menemukan manajer kunci eksternal dekat dengan Wilayah AWS sehingga waktu pulang-pergi jaringan (RTT) adalah 35 milidetik atau kurang.

Jika proxy penyimpanan kunci eksternal Anda mengizinkannya, Anda dapat mengubah pengelola kunci eksternal yang Anda kaitkan dengan proxy penyimpanan kunci eksternal Anda, tetapi manajer kunci eksternal yang baru harus berupa cadangan atau snapshot dengan materi kunci yang sama. Jika kunci eksternal yang Anda kaitkan dengan kunci KMS tidak lagi tersedia untuk proxy penyimpanan kunci eksternal Anda, tidak AWS KMS dapat mendekripsi ciphertext yang dienkripsi dengan kunci KMS.

Manajer kunci eksternal harus dapat diakses oleh proxy penyimpanan kunci eksternal. Jika GetHealthStatusrespons dari proxy melaporkan bahwa semua instance pengelola kunci eksternal

adalah`Unavailable`, semua upaya untuk membuat penyimpanan kunci eksternal gagal dengan file. `XksProxyUriUnreachableException`

Proksi penyimpanan kunci eksternal

Anda harus menentukan proxy penyimpanan kunci eksternal (proxy XKS) yang sesuai dengan persyaratan desain dalam Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal. Anda dapat mengembangkan atau membeli proxy penyimpanan kunci eksternal, atau menggunakan proxy penyimpanan kunci eksternal yang disediakan oleh atau dibangun ke dalam manajer kunci eksternal Anda. AWS KMSmerekomendasikan agar proxy penyimpanan kunci eksternal Anda dikonfigurasi untuk menangani hingga 1800 permintaan per detik dan merespons dalam batas waktu 250 milidetik untuk setiap permintaan. Kami menyarankan Anda menemukan manajer kunci eksternal dekat dengan Wilayah AWS sehingga waktu pulang-pergi jaringan (RTT) adalah 35 milidetik atau kurang.

Anda dapat menggunakan proxy penyimpanan kunci eksternal untuk lebih dari satu penyimpanan kunci eksternal, tetapi setiap penyimpanan kunci eksternal harus memiliki titik akhir dan jalur URI yang unik dalam proxy penyimpanan kunci eksternal untuk permintaannya.

Jika Anda menggunakan konektivitas layanan titik akhir VPC, Anda dapat menemukan proxy penyimpanan kunci eksternal di VPC Amazon Anda, tetapi itu tidak diperlukan. Anda dapat menemukan proxy Anda di luarAWS, seperti di pusat data pribadi Anda, dan menggunakan layanan titik akhir VPC hanya untuk berkomunikasi dengan proxy.

Kredensi otentikasi proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan kredensi otentikasi proxy penyimpanan kunci eksternal Anda ()`XksProxyAuthenticationCredential`.

Anda harus membuat kredensi otentikasi (`XksProxyAuthenticationCredential`) untuk AWS KMS proxy penyimpanan kunci eksternal Anda. AWS KMSmengautentikasi proxy Anda dengan menandatangani permintaannya menggunakan proses Signature Version 4 (SigV4) dengan kredensi otentikasi proxy penyimpanan kunci eksternal. Anda menentukan kredensi otentikasi saat membuat penyimpanan kunci eksternal dan Anda dapat mengubahnya kapan saja. Jika proxy Anda memutar kredensi Anda, pastikan untuk memperbarui nilai kredensi untuk penyimpanan kunci eksternal Anda.

Kredensi otentikasi proxy memiliki dua bagian. Anda harus menyediakan kedua bagian untuk toko kunci eksternal Anda.

- ID kunci akses: Mengidentifikasi kunci akses rahasia. Anda dapat memberikan ID ini dalam teks biasa.

- Kunci akses rahasia: Bagian rahasia dari kredensi. AWS KMSmengenkripsi kunci akses rahasia di kredensi sebelum menyimpannya.

Kredensyal SigV4 yang AWS KMS digunakan untuk menandatangani permintaan ke proxy penyimpanan kunci eksternal tidak terkait dengan kredensyal SigV4 apa pun yang terkait dengan prinsip apa pun di akun Anda. AWS Identity and Access Management AWS Jangan gunakan kembali kredensi IAM SiGv4 apa pun untuk proxy penyimpanan kunci eksternal Anda.

Konektivitas proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan opsi konektivitas proxy penyimpanan kunci eksternal Anda (`XksProxyConnectivity`).

AWS KMSdapat berkomunikasi dengan proxy penyimpanan kunci eksternal Anda dengan menggunakan titik akhir publik atau layanan titik akhir Amazon Virtual Private Cloud (Amazon VPC). Meskipun titik akhir publik lebih mudah untuk dikonfigurasi dan dipelihara, itu mungkin tidak memenuhi persyaratan keamanan untuk setiap instalasi. Jika Anda memilih opsi konektivitas layanan titik akhir VPC Amazon, Anda harus membuat dan memelihara komponen yang diperlukan, termasuk VPC Amazon dengan setidaknya dua subnet di dua Availability Zone yang berbeda, layanan titik akhir VPC dengan penyeimbang beban jaringan dan grup target, dan nama DNS pribadi untuk layanan titik akhir VPC.

Anda dapat mengubah opsi konektivitas proxy untuk penyimpanan kunci eksternal Anda. Namun, Anda harus memastikan bahwa ketersediaan berkelanjutan dari materi utama yang terkait dengan kunci KMS di toko kunci eksternal Anda. Jika tidak, AWS KMS tidak dapat mendekripsi ciphertext apa pun yang dienkripsi dengan kunci KMS tersebut.

Untuk bantuan menentukan opsi konektivitas proxy mana yang terbaik untuk penyimpanan kunci eksternal Anda, lihatMemilih opsi konektivitas proxy. Untuk bantuan membuat konfigurasi konektivitas layanan titik akhir VPC, lihat. Mengkonfigurasi konektivitas layanan titik akhir VPC

Titik akhir URI proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan endpoint (`XksProxyUriEndpoint`) yang AWS KMS digunakan untuk mengirim permintaan ke proxy penyimpanan kunci eksternal.

Protokol harus HTTPS. AWS KMSberkomunikasi di port 443. Jangan tentukan port dalam nilai titik akhir URI proxy.

- Konektivitas titik akhir publik - Tentukan titik akhir yang tersedia untuk umum untuk proxy penyimpanan kunci eksternal Anda. Titik akhir ini harus dapat dijangkau sebelum Anda membuat penyimpanan kunci eksternal Anda.
- Konektivitas layanan titik akhir VPC — Tentukan `https://` diikuti dengan nama DNS pribadi dari layanan titik akhir VPC.

Sertifikat server TLS yang dikonfigurasi pada proxy penyimpanan kunci eksternal harus cocok dengan nama domain di titik akhir URI proxy penyimpanan kunci eksternal dan dikeluarkan oleh otoritas sertifikat yang didukung untuk penyimpanan kunci eksternal. Untuk daftar, lihat Otoritas Sertifikat Tepercaya. Otoritas sertifikat Anda akan memerlukan bukti kepemilikan domain sebelum menerbitkan sertifikat TLS.

Nama umum subjek (CN) pada sertifikat TLS harus cocok dengan nama DNS pribadi. Misalnya, jika nama DNS pribadi adalah`myproxy-private.xks.example.com`, CN pada sertifikat TLS harus atau. `myproxy-private.xks.example.com` `*.xks.example.com`

Anda dapat mengubah titik akhir URI proxy Anda, tetapi pastikan bahwa proxy penyimpanan kunci eksternal memiliki akses ke materi kunci yang terkait dengan kunci KMS di penyimpanan kunci eksternal Anda. Jika tidak, AWS KMS tidak dapat mendekripsi ciphertext apa pun yang dienkripsi dengan kunci KMS tersebut.

Persyaratan keunikan

- Nilai endpoint (`XksProxyUriEndpoint`) dan proxy URI path () gabungan proxy URI (`XksProxyUriPath`) harus unik di Akun AWS dan Region.
- Penyimpanan kunci eksternal dengan konektivitas titik akhir publik dapat berbagi titik akhir URI proxy yang sama, asalkan memiliki nilai jalur URI proxy yang berbeda.
- Penyimpanan kunci eksternal dengan konektivitas titik akhir publik tidak dapat menggunakan nilai titik akhir URI proxy yang sama dengan penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC yang samaWilayah AWS, meskipun penyimpanan kunci berbeda. Akun AWS
- Setiap penyimpanan kunci eksternal dengan konektivitas titik akhir VPC harus memiliki nama DNS pribadinya sendiri. Titik akhir URI proxy (nama DNS pribadi) harus unik di Akun AWS dan Wilayah.

Jalur URI proxy

Untuk membuat penyimpanan kunci eksternal, Anda harus menentukan jalur dasar di proxy penyimpanan kunci eksternal ke API proxy yang diperlukan. Nilai harus dimulai dengan / dan

harus diakhiri dengan /kms/xks/v1 di mana `v1` mewakili versi API untuk proxy penyimpanan kunci eksternal. AWS KMS Jalur ini dapat menyertakan awalan opsional antara elemen yang diperlukan seperti`/example-prefix/kms/xks/v1`. Untuk menemukan nilai ini, lihat dokumentasi untuk proxy penyimpanan kunci eksternal Anda.

AWS KMSmengirimkan permintaan proxy ke alamat yang ditentukan oleh penggabungan titik akhir URI proxy dan jalur URI proxy. Misalnya, jika titik akhir URI proxy `https://myproxy.xks.example.com` dan jalur URI proxy adalah`/kms/xks/v1`, AWS KMS kirimkan permintaan API proksi ke`https://myproxy.xks.example.com/kms/xks/v1`.

Anda dapat mengubah jalur URI proxy Anda, tetapi pastikan bahwa proxy penyimpanan kunci eksternal memiliki akses ke materi kunci yang terkait dengan kunci KMS di penyimpanan kunci eksternal Anda. Jika tidak, AWS KMS tidak dapat mendekripsi ciphertext apa pun yang dienkripsi dengan kunci KMS tersebut.

Persyaratan keunikan

- Nilai endpoint (`XksProxyUriEndpoint`) dan proxy URI path () gabungan proxy URI (`XksProxyUriPath`) harus unik di Akun AWS dan Region.

Layanan titik akhir VPC

Menentukan nama layanan endpoint Amazon VPC yang digunakan untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda. Komponen ini hanya diperlukan untuk penyimpanan kunci eksternal yang menggunakan konektivitas layanan titik akhir VPC. Untuk bantuan menyiapkan dan mengonfigurasi layanan titik akhir VPC Anda untuk penyimpanan kunci eksternal, lihat. Mengkonfigurasi konektivitas layanan titik akhir VPC

Layanan titik akhir VPC harus memiliki properti berikut:

- Layanan titik akhir VPC harus sama Akun AWS dan Wilayah sebagai penyimpanan kunci eksternal.
- Ini harus memiliki penyeimbang beban jaringan (NLB) yang terhubung ke setidaknya dua subnet, masing-masing di Availability Zone yang berbeda.
- Daftar prinsip izin untuk layanan titik akhir VPC harus menyertakan prinsip AWS KMS layanan untuk Wilayah:, seperti. `cks.kms.<region>.amazonaws.com` `cks.kms.us-east-1.amazonaws.com`
- Itu tidak boleh memerlukan penerimaan permintaan koneksi.

- Itu harus memiliki nama DNS pribadi dalam domain publik tingkat yang lebih tinggi. Misalnya, Anda dapat memiliki nama DNS pribadi myproxy-private.xks.example.com di domain publik. `xks.example.com`

  Nama DNS pribadi untuk penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC harus unik di dalamnya. Wilayah AWS
- Status verifikasi domain domain nama DNS pribadi harus`verified`.
- Sertifikat server TLS yang dikonfigurasi pada proxy penyimpanan kunci eksternal harus menentukan nama host DNS pribadi di mana titik akhir dapat dijangkau.

Persyaratan keunikan

- Toko kunci eksternal dengan konektivitas titik akhir VPC dapat berbagi`Amazon VPC`, tetapi setiap toko kunci eksternal harus memiliki layanan titik akhir VPC sendiri dan nama DNS pribadi.

File konfigurasi proxy

File konfigurasi proxy adalah file berbasis JSON opsional yang berisi nilai untuk jalur URI proxy dan properti kredensi otentikasi proxy dari penyimpanan kunci eksternal Anda. Saat membuat atau mengedit penyimpanan kunci eksternal di AWS KMS konsol, Anda dapat mengunggah file konfigurasi proxy untuk menyediakan nilai konfigurasi untuk penyimpanan kunci eksternal Anda. Menggunakan file ini menghindari kesalahan pengetikan dan penyisipan, dan memastikan bahwa nilai di penyimpanan kunci eksternal Anda cocok dengan nilai di proxy penyimpanan kunci eksternal Anda.

File konfigurasi proxy dihasilkan oleh proxy penyimpanan kunci eksternal. Untuk mengetahui apakah proxy penyimpanan kunci eksternal Anda menawarkan file konfigurasi proxy, lihat dokumentasi proxy penyimpanan kunci eksternal Anda.

Berikut ini adalah contoh file konfigurasi proxy yang terbentuk dengan baik dengan nilai fiktif.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
  }
}
```

Anda dapat mengunggah file konfigurasi proxy hanya saat membuat atau mengedit penyimpanan kunci eksternal di AWS KMS konsol. Anda tidak dapat menggunakannya dengan UpdateCustomKeyStoreoperasi CreateCustomKeyStoreatau, tetapi Anda dapat menggunakan nilai dalam file konfigurasi proxy untuk memastikan bahwa nilai parameter Anda benar.

Buat toko kunci eksternal (konsol)

Sebelum membuat penyimpanan kunci eksternal, tinjauMerencanakan toko kunci eksternal, pilih jenis konektivitas proxy Anda, dan pastikan bahwa Anda telah membuat dan mengonfigurasi semua komponen yang diperlukan. Jika Anda memerlukan bantuan untuk menemukan salah satu nilai yang diperlukan, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau perangkat lunak manajemen kunci Anda.

> **Note**
>
> Saat Anda membuat penyimpanan kunci eksternal diAWS Management Console, Anda dapat mengunggah file konfigurasi proxy berbasis JSON dengan nilai untuk jalur URI proxy dan kredensi otentikasi proxy. Beberapa proxy menghasilkan file ini untuk Anda. Hal ini tidak diperlukan.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Pilih Buat toko kunci eksternal.
5. Masukkan nama ramah untuk toko kunci eksternal. Nama harus unik di antara semua toko kunci eksternal di akun Anda.

   > **Important**
   >
   > Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

6. Pilih jenis konektivitas proxy Anda.

Pilihan konektivitas proxy Anda menentukan komponen yang diperlukan untuk proxy penyimpanan kunci eksternal Anda. Untuk bantuan membuat pilihan ini, lihatMemilih opsi konektivitas proxy.

7. Pilih atau masukkan nama layanan titik akhir VPC untuk penyimpanan kunci eksternal ini. Langkah ini hanya muncul ketika jenis konektivitas proxy penyimpanan kunci eksternal Anda adalah layanan titik akhir VPC.

   Layanan titik akhir VPC dan VPC-nya harus memenuhi persyaratan untuk penyimpanan kunci eksternal. Untuk detailnya, lihat the section called “Memasang prasyarat”.

8. Masukkan titik akhir URI proxy Anda. Protokol harus HTTPS. AWS KMSberkomunikasi di port 443. Jangan tentukan port dalam nilai titik akhir URI proxy.

   Jika AWS KMS mengenali layanan titik akhir VPC yang Anda tentukan pada langkah sebelumnya, itu melengkapi bidang ini untuk Anda.

   Untuk konektivitas titik akhir publik, masukkan URI titik akhir yang tersedia untuk umum. Untuk konektivitas titik akhir VPC, masukkan `https://` diikuti dengan nama DNS pribadi dari layanan titik akhir VPC.

9. Untuk memasukkan nilai untuk awalan jalur URI proxy dan kredensi otentikasi proxy, unggah file konfigurasi proxy, atau masukkan nilai secara manual.

   - Jika Anda memiliki file konfigurasi proxy opsional yang berisi nilai untuk jalur URI proxy dan kredensi otentikasi proxy, pilih Unggah file konfigurasi. Ikuti langkah-langkah untuk mengunggah file.

     Saat file diunggah, konsol menampilkan nilai dari file di bidang yang dapat diedit. Anda dapat mengubah nilai sekarang atau mengedit nilai-nilai ini setelah penyimpanan kunci eksternal dibuat.

     Untuk menampilkan nilai kunci akses rahasia, pilih Tampilkan kunci akses rahasia.

   - Jika Anda tidak memiliki file konfigurasi proxy, Anda dapat memasukkan jalur URI proxy dan nilai kredensi otentikasi proxy secara manual.

     a. Jika Anda tidak memiliki file konfigurasi proxy, Anda dapat memasukkan URI proxy secara manual. Konsol menyediakan nilai /kms/xks/v1 yang diperlukan.

Jika jalur URI proxy Anda menyertakan awalan opsional, seperti `example-prefix` in`/example-prefix/kms/xks/v1`, masukkan awalan di bidang awalan jalur Proxy URI. Jika tidak, biarkan bidang kosong.

b. Jika Anda tidak memiliki file konfigurasi proxy, Anda dapat memasukkan kredensi otentikasi proxy Anda secara manual. Baik ID kunci akses dan kunci akses rahasia diperlukan.

   - Dalam kredensi proxy: ID kunci akses, masukkan ID kunci akses dari kredensi otentikasi proxy. ID kunci akses mengidentifikasi kunci akses rahasia.
   - Di Proxy credential: Secret Access Key, masukkan kunci akses rahasia dari kredensi otentikasi proxy.

   Untuk menampilkan nilai kunci akses rahasia, pilih Tampilkan kunci akses rahasia.

   Prosedur ini tidak mengatur atau mengubah kredensi otentikasi yang Anda buat pada proxy penyimpanan kunci eksternal Anda. Itu hanya mengaitkan nilai-nilai ini dengan toko kunci eksternal Anda. Untuk informasi tentang pengaturan, perubahan, dan kredensi autentikasi proxy yang berputar, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau perangkat lunak manajemen kunci.

   Jika kredensi otentikasi proxy Anda berubah, edit setelan kredensyal untuk penyimpanan kunci eksternal Anda.

10. Pilih Buat toko kunci eksternal.

Ketika prosedur berhasil, toko kunci eksternal baru muncul dalam daftar toko kunci eksternal di akun dan wilayah. Jika prosedur gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat CreateKey kesalahan untuk kunci eksternal.

Berikutnya: Toko kunci eksternal baru tidak terhubung secara otomatis. Sebelum Anda dapat membuat AWS KMS keys di toko kunci eksternal Anda, Anda harus menghubungkan toko kunci eksternal ke proxy penyimpanan kunci eksternal.

Buat toko kunci eksternal (API)

Anda dapat menggunakan CreateCustomKeyStoreoperasi untuk membuat toko kunci eksternal baru. Untuk bantuan menemukan nilai untuk parameter yang diperlukan, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau perangkat lunak manajemen kunci.

> **Tip**
>
> Anda tidak dapat mengunggah file konfigurasi proxy saat menggunakan `CreateCustomKeyStore` operasi. Namun, Anda dapat menggunakan nilai dalam file konfigurasi proxy untuk memastikan bahwa nilai parameter Anda benar.

Untuk membuat penyimpanan kunci eksternal, `CreateCustomKeyStore` operasi memerlukan nilai parameter berikut.

- `CustomKeyStoreName`— Nama ramah untuk toko kunci eksternal yang unik di akun.

  > **Important**
  >
  > Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.

- `CustomKeyStoreType`— Tentukan`EXTERNAL_KEY_STORE`.
- `XksProxyConnectivity`— Tentukan `PUBLIC_ENDPOINT` atau`VPC_ENDPOINT_SERVICE`.
- `XksProxyAuthenticationCredential`— Tentukan ID kunci akses dan kunci akses rahasia.
- `XksProxyUriEndpoint`— Titik akhir yang AWS KMS digunakan untuk berkomunikasi dengan proxy penyimpanan kunci eksternal Anda.
- `XksProxyUriPath`— Jalur dalam proxy ke API proxy.
- `XksProxyVpcEndpointServiceName`— Diperlukan hanya ketika `XksProxyConnectivity` nilai Anda`VPC_ENDPOINT_SERVICE`.

> **Note**
>
> Jika Anda menggunakan AWS CLI versi 1.0, jalankan perintah berikut sebelum menentukan parameter dengan nilai HTTP atau HTTPS, seperti `XksProxyUriEndpoint` parameter.

```
aws configure set cli_follow_urlparam false
```

Jika tidak, AWS CLI versi 1.0 menggantikan nilai parameter dengan konten yang ditemukan di alamat URI tersebut, menyebabkan kesalahan berikut:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

Contoh berikut menggunakan nilai fiktif. Sebelum menjalankan perintah, ganti dengan nilai yang valid untuk penyimpanan kunci eksternal Anda.

Buat toko kunci eksternal dengan konektivitas titik akhir publik.

```
$ 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>
```

Buat toko kunci eksternal dengan konektivitas layanan titik akhir VPC.

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStoreVPC \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
        --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-svc-
example \
        --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

Saat operasi berhasil, `CreateCustomKeyStore` mengembalikan ID penyimpanan kunci kustom, seperti yang ditunjukkan dalam contoh tanggapan berikut.

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Jika operasi gagal, perbaiki kesalahan yang ditunjukkan oleh pengecualian, dan coba lagi. Untuk bantuan tambahan, lihat Memecahkan masalah toko kunci eksternal.

Berikutnya: Untuk menggunakan penyimpanan kunci eksternal, sambungkan ke proxy penyimpanan kunci eksternal.

Mengedit properti penyimpanan kunci eksternal

Anda dapat mengedit properti yang dipilih dari penyimpanan kunci eksternal yang ada.

Anda dapat mengedit beberapa properti saat penyimpanan kunci eksternal terhubung atau terputus. Untuk properti lain, Anda harus terlebih dahulu memutuskan penyimpanan kunci eksternal Anda dari proxy penyimpanan kunci eksternal. Status koneksi penyimpanan kunci eksternal harus`DISCONNECTED`. Sementara toko kunci eksternal Anda terputus, Anda dapat mengelola penyimpanan kunci dan kunci KMS-nya, tetapi Anda tidak dapat membuat atau menggunakan kunci KMS di toko kunci eksternal. Untuk menemukan status koneksi penyimpanan kunci eksternal Anda, gunakan DescribeCustomKeyStoresoperasi atau lihat bagian Konfigurasi umum pada halaman detail untuk penyimpanan kunci eksternal.

Sebelum memperbarui properti penyimpanan kunci eksternal Anda, AWS KMS kirimkan GetHealthStatuspermintaan ke proxy penyimpanan kunci eksternal menggunakan nilai baru. Jika permintaan berhasil, ini menunjukkan bahwa Anda dapat menghubungkan dan mengautentikasi ke proxy penyimpanan kunci eksternal dengan nilai properti yang diperbarui. Jika permintaan gagal, operasi edit gagal dengan pengecualian yang mengidentifikasi kesalahan.

Saat operasi edit selesai, nilai properti yang diperbarui untuk penyimpanan kunci eksternal Anda akan muncul di AWS KMS konsol dan `DescribeCustomKeyStores` respons. Namun, perlu waktu hingga lima menit agar perubahan sepenuhnya efektif.

Jika Anda mengedit penyimpanan kunci eksternal di AWS KMS konsol, Anda memiliki opsi untuk mengunggah file konfigurasi proxy berbasis JSON yang menentukan jalur URI proxydan kredensi otentikasi proxy. Beberapa proxy penyimpanan kunci eksternal menghasilkan file ini untuk Anda. Untuk detailnya, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau pengelola kunci eksternal.

> **Warning**
>
> Nilai properti yang diperbarui harus menghubungkan penyimpanan kunci eksternal Anda ke proxy untuk pengelola kunci eksternal yang sama dengan nilai sebelumnya, atau untuk cadangan atau snapshot dari pengelola kunci eksternal dengan kunci kriptografi yang sama. Jika penyimpanan kunci eksternal Anda secara permanen kehilangan aksesnya ke kunci eksternal yang terkait dengan kunci KMS-nya, ciphertext yang dienkripsi di bawah kunci eksternal tersebut tidak dapat dipulihkan. Secara khusus, mengubah konektivitas proxy dari penyimpanan kunci eksternal dapat AWS KMS mencegah mengakses kunci eksternal Anda.

> **Tip**
>
> Beberapa manajer kunci eksternal menyediakan metode yang lebih sederhana untuk mengedit properti penyimpanan kunci eksternal. Untuk detailnya, lihat dokumentasi pengelola kunci eksternal Anda.

Anda dapat mengubah properti berikut dari penyimpanan kunci eksternal.

Properti penyimpanan kunci eksternal yang dapat diedit	Status koneksi apa pun	Memerlukan status Terputus
Nama penyimpanan kunci kustom Nama ramah yang diperlukan untuk toko kunci khusus. **Important** Jangan sertakan informasi rahasia atau sensitif di bidang ini. Bidang ini dapat ditampilkan dalam plaintext di CloudTrail log dan output lainnya.	✓	
Kredensi otentikasi proxy () XksProxyAuthenticationCredential	✓	

Properti penyimpanan kunci eksternal yang dapat diedit	Status koneksi apa pun	Memerlukan status Terputus
(Anda harus menentukan ID kunci akses dan kunci akses rahasia, bahkan jika Anda hanya mengubah satu elemen.)		
Jalur URI proxy (XksProxyUriPath)	✓	
Konektivitas proxy (XksProxyConnectivity) (Anda juga harus memperbarui titik akhir URI proxy. Jika Anda mengubah ke konektivitas layanan titik akhir VPC, Anda harus menentukan nama layanan titik akhir VPC proxy.)		✓
Titik akhir URI proxy () XksProxyUriEndpoint Jika Anda mengubah URI titik akhir proxy, Anda mungkin juga perlu mengubah sertifikat TLS terkait.		✓
Nama layanan titik akhir VPC proxy () XksProxyVpcEndpointServiceName (Bidang ini diperlukan untuk konektivitas layanan titik akhir VPC)		✓

Topik

- Mengedit penyimpanan kunci eksternal (konsol)
- Mengedit penyimpanan kunci eksternal (API)

Mengedit penyimpanan kunci eksternal (konsol)

Saat Anda mengedit penyimpanan kunci, Anda dapat mengubah salah satu atau nilai yang dapat diedit. Beberapa perubahan mengharuskan penyimpanan kunci eksternal terputus dari proxy penyimpanan kunci eksternal.

Jika Anda mengedit jalur URI proxy atau kredensi otentikasi proxy, Anda dapat memasukkan nilai baru atau mengunggah file konfigurasi proxy penyimpanan kunci eksternal yang menyertakan nilai baru.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Pilih baris toko kunci eksternal yang ingin Anda edit.
5. Jika perlu, lepaskan penyimpanan kunci eksternal dari proxy penyimpanan kunci eksternal. Dari menu Key Store Actions, pilih Disconnect.
6. Dari menu Key store actions, pilih Edit.
7. Ubah satu atau beberapa properti penyimpanan kunci eksternal yang dapat diedit. Anda juga dapat mengunggah file konfigurasi proxy penyimpanan kunci eksternal dengan nilai untuk jalur URI proxy dan kredensil otentikasi proksi. Anda dapat menggunakan file konfigurasi proxy meskipun beberapa nilai yang ditentukan dalam file tidak berubah.
8. Pilih Perbarui toko kunci eksternal.
9. Tinjau peringatan, dan jika Anda memutuskan untuk melanjutkan, konfirmasikan peringatan, lalu pilih Perbarui toko kunci eksternal.

   Ketika prosedur berhasil, pesan menjelaskan properti yang Anda edit. Ketika prosedur gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya.
10. Jika perlu, sambungkan kembali toko kunci eksternal. Dari menu Key Store Actions, pilih Connect.

    Anda dapat membiarkan toko kunci eksternal terputus. Tetapi saat terputus, Anda tidak dapat membuat kunci KMS di penyimpanan kunci eksternal atau menggunakan kunci KMS di penyimpanan kunci eksternal dalam operasi kriptografi.

Mengedit penyimpanan kunci eksternal (API)

Untuk mengubah properti penyimpanan kunci eksternal, gunakan UpdateCustomKeyStoreoperasi. Anda dapat mengubah beberapa properti penyimpanan kunci eksternal dalam operasi yang sama. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti.

Gunakan `CustomKeyStoreId` parameter untuk mengidentifikasi penyimpanan kunci eksternal. Gunakan parameter lain untuk mengubah properti. Anda tidak dapat menggunakan file konfigurasi proxy dengan `UpdateCustomKeyStore` operasi. File konfigurasi proxy hanya didukung oleh AWS KMS konsol. Namun, Anda dapat menggunakan file konfigurasi proxy untuk membantu Anda menentukan nilai parameter yang benar untuk proxy penyimpanan kunci eksternal Anda.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Sebelum Anda mulai, jika perlu, lepaskan penyimpanan kunci eksternal dari proxy penyimpanan kunci eksternal. Setelah memperbarui, jika perlu, Anda dapat menghubungkan kembali toko kunci eksternal ke proxy penyimpanan kunci eksternal. Anda dapat meninggalkan penyimpanan kunci eksternal dalam keadaan terputus, tetapi Anda harus menghubungkannya kembali sebelum Anda dapat membuat kunci KMS baru di toko kunci atau menggunakan kunci KMS yang ada di toko kunci untuk operasi kriptografi.

> **Note**
>
> Jika Anda menggunakan AWS CLI versi 1.0, jalankan perintah berikut sebelum menentukan parameter dengan nilai HTTP atau HTTPS, seperti `XksProxyUriEndpoint` parameter.
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> Jika tidak, AWS CLI versi 1.0 menggantikan nilai parameter dengan konten yang ditemukan di alamat URI tersebut, menyebabkan kesalahan berikut:
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

Ubah nama toko kunci eksternal

Contoh pertama menggunakan UpdateCustomKeyStoreoperasi untuk mengubah nama ramah dari penyimpanan kunci eksternal menjadi`XksKeyStore`. Perintah menggunakan parameter `CustomKeyStoreId` untuk mengidentifikasi penyimpanan kunci kustom dan `CustomKeyStoreName` untuk menentukan nama baru untuk penyimpanan kunci kustom. Ganti semua nilai contoh dengan nilai aktual untuk penyimpanan kunci eksternal Anda.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name XksKeyStore
```

Mengubah kredensi otentikasi proxy

Contoh berikut memperbarui kredensi otentikasi proxy yang AWS KMS digunakan untuk mengautentikasi ke proxy penyimpanan kunci eksternal. Anda dapat menggunakan perintah seperti ini untuk memperbarui kredensi jika diputar pada proxy Anda.

Perbarui kredensi pada proxy penyimpanan kunci eksternal Anda terlebih dahulu. Kemudian gunakan fitur ini untuk melaporkan perubahan keAWS KMS. (Proxy Anda akan secara singkat mendukung kredensi lama dan baru sehingga Anda punya waktu untuk memperbarui kredensi Anda.) AWS KMS

Anda harus selalu menentukan ID kunci akses dan kunci akses rahasia di kredensi, meskipun hanya satu nilai yang diubah.

Dua perintah pertama mengatur variabel untuk menyimpan nilai kredensi. `UpdateCustomKeyStore`Operasi menggunakan `CustomKeyStoreId` parameter untuk mengidentifikasi penyimpanan kunci eksternal. Ini menggunakan `XksProxyAuthenticationCredential` parameter dengan `AccessKeyId` dan `RawSecretAccessKey` bidangnya untuk menentukan kredensi baru. Ganti semua nilai contoh dengan nilai aktual untuk penyimpanan kunci eksternal Anda.

```
$ 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
```

Ubah jalur URI proxy

Contoh berikut memperbarui jalur URI proxy (`XksProxyUriPath`). Kombinasi titik akhir URI proxy dan jalur URI proxy harus unik di Akun AWS dan Wilayah. Ganti semua nilai contoh dengan nilai aktual untuk penyimpanan kunci eksternal Anda.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-uri-path /kms/xks/v1
```

Ubah ke konektivitas layanan titik akhir VPC

Contoh berikut menggunakan UpdateCustomKeyStoreoperasi untuk mengubah jenis konektivitas proxy penyimpanan kunci eksternal ke`VPC_ENDPOINT_SERVICE`. Untuk membuat perubahan ini, Anda harus menentukan nilai yang diperlukan untuk konektivitas layanan titik akhir VPC, termasuk nama layanan titik akhir VPC (`XksProxyVpcEndpointServiceName`) dan nilai titik akhir URI proxy () yang menyertakan nama DNS pribadi untuk layanan titik akhir VPC. `XksProxyUriEndpoint` Ganti semua nilai contoh dengan nilai aktual untuk penyimpanan kunci eksternal Anda.

```
$ 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
```

Ubah ke konektivitas titik akhir publik

Contoh berikut mengubah jenis konektivitas proxy penyimpanan kunci eksternal menjadi`PUBLIC_ENDPOINT`. Ketika Anda membuat perubahan ini, Anda harus memperbarui nilai endpoint (`XksProxyUriEndpoint`) URI proxy. Ganti semua nilai contoh dengan nilai aktual untuk penyimpanan kunci eksternal Anda.

> **Note**
>
> Konektivitas titik akhir VPC memberikan keamanan yang lebih besar daripada konektivitas titik akhir publik. Sebelum beralih ke konektivitas titik akhir publik, pertimbangkan opsi lain, termasuk menemukan proxy penyimpanan kunci eksternal Anda di tempat dan menggunakan VPC hanya untuk komunikasi.

```
$ 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
```

Melihat toko kunci eksternal

Anda dapat melihat toko kunci eksternal di setiap akun dan Wilayah dengan menggunakan AWS KMS konsol atau dengan menggunakan DescribeCustomKeyStoresoperasi.

Ketika Anda melihat penyimpanan kunci eksternal, Anda dapat melihat yang berikut:

- Informasi dasar tentang toko kunci, termasuk nama ramah, ID, jenis toko kunci, dan tanggal pembuatannya.
- Informasi konfigurasi untuk proxy penyimpanan kunci eksternal, termasuk jenis konektivitas, titik akhir dan jalur URI proxy, dan ID kunci akses kredensi otentikasi proxy Anda saat ini.
- Jika proxy penyimpanan kunci eksternal menggunakan konektivitas layanan titik akhir VPC, konsol akan menampilkan nama layanan titik akhir VPC.
- Keadaan koneksi saat ini.

> **Note**
>
> Nilai status koneksi Terputus menunjukkan bahwa penyimpanan kunci eksternal tidak pernah terhubung, atau sengaja terputus dari proxy penyimpanan kunci eksternalnya. Namun, jika upaya Anda untuk menggunakan kunci KMS di penyimpanan kunci eksternal yang terhubung gagal, itu mungkin menunjukkan masalah dengan penyimpanan kunci eksternal atau proksi. Untuk bantuan, lihat Kesalahan koneksi penyimpanan kunci eksternal.

- Bagian Pemantauan dengan grafik CloudWatch metrik Amazon yang dirancang untuk membantu Anda mendeteksi dan menyelesaikan masalah dengan penyimpanan kunci eksternal Anda. Untuk bantuan menafsirkan grafik, menggunakannya dalam perencanaan dan pemecahan masalah, dan membuat CloudWatch alarm berdasarkan metrik dalam grafik, lihat. Memantau toko kunci eksternal

Lihat juga:

- Melihat tombol KMS di toko kunci eksternal
- Logging panggilan AWS KMS API dengan AWS CloudTrail

Topik

- Properti penyimpanan kunci eksternal
- Lihat toko kunci eksternal (konsol)
- Melihat penyimpanan kunci eksternal (API)

Properti penyimpanan kunci eksternal

Properti berikut dari penyimpanan kunci eksternal terlihat di AWS KMS konsol dan DescribeCustomKeyStoresresponsnya.

Properti toko kunci kustom

Nilai berikut muncul di bagian konfigurasi umum halaman detail untuk setiap toko kunci kustom. Properti ini berlaku untuk semua toko kunci kustom, termasuk toko kunci dan toko AWS CloudHSM kunci eksternal.

ID penyimpanan kunci kustom

ID unik yang ditetapkan AWS KMS ke toko kunci kustom.

Nama penyimpanan kunci kustom

Nama ramah yang Anda tetapkan ke toko kunci kustom saat Anda membuatnya. Anda dapat mengubah nilai ini kapan saja.

Jenis toko kunci khusus

Jenis toko kunci khusus. Nilai yang valid adalah AWS CloudHSM (`AWS_CLOUDHSM`) atau External key store (`EXTERNAL_KEY_STORE`). Anda tidak dapat mengubah jenis setelah Anda membuat toko kunci kustom.

Tanggal pembuatan

Tanggal penyimpanan kunci khusus dibuat. Tanggal ini ditampilkan dalam waktu setempat untukWilayah AWS.

Status koneksi

Menunjukkan apakah toko kunci khusus terhubung ke toko kunci pendukungnya. Status koneksi `DISCONNECTED` hanya jika toko kunci khusus tidak pernah terhubung ke toko kunci pendukungnya, atau sengaja terputus. Untuk detailnya, lihat the section called “Status koneksi”.

Properti konfigurasi penyimpanan kunci eksternal

Nilai-nilai berikut muncul di bagian konfigurasi proxy penyimpanan kunci eksternal dari halaman detail untuk setiap penyimpanan kunci eksternal dan dalam `XksProxyConfiguration` elemen DescribeCustomKeyStoresrespons. Untuk penjelasan rinci tentang setiap bidang, termasuk persyaratan keunikan dan bantuan menentukan nilai yang benar untuk setiap bidang, lihat the section called “Memasang prasyarat” di topik Membuat penyimpanan kunci eksternal.

Konektivitas proxy

Menunjukkan apakah penyimpanan kunci eksternal menggunakan konektivitas titik akhir publik atau konektivitas layanan titik akhir VPC.

Titik akhir URI proxy

Titik akhir yang AWS KMS digunakan untuk terhubung ke proxy penyimpanan kunci eksternal Anda.

Jalur URI proxy

Jalur dari titik akhir URI proxy tempat AWS KMS mengirimkan permintaan API proxy.

Kredensi proxy: ID kunci akses

Bagian dari kredensi otentikasi proxy yang Anda buat di proxy penyimpanan kunci eksternal Anda. ID kunci akses mengidentifikasi kunci akses rahasia dalam kredensi.

AWS KMSmenggunakan proses penandatanganan SiGv4 dan kredensi otentikasi proxy untuk menandatangani permintaannya ke proxy penyimpanan kunci eksternal Anda. Kredensi dalam tanda tangan memungkinkan proxy penyimpanan kunci eksternal untuk mengautentikasi permintaan atas nama Anda. AWS KMS

Nama layanan titik akhir VPC

Nama layanan endpoint Amazon VPC yang mendukung toko kunci eksternal Anda. Nilai ini hanya muncul ketika penyimpanan kunci eksternal menggunakan konektivitas layanan titik akhir VPC. Anda dapat menemukan proxy penyimpanan kunci eksternal Anda di VPC atau menggunakan layanan titik akhir VPC untuk berkomunikasi secara aman dengan proxy penyimpanan kunci eksternal Anda.

Lihat toko kunci eksternal (konsol)

Untuk melihat penyimpanan kunci eksternal di akun dan Wilayah tertentu, gunakan prosedur berikut.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Untuk melihat informasi rinci tentang toko kunci eksternal, pilih nama toko kunci.

Melihat penyimpanan kunci eksternal (API)

Untuk melihat toko kunci eksternal Anda, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah. Namun Anda dapat menggunakan parameter `CustomKeyStoreId` atau `CustomKeyStoreName` (tetapi tidak keduanya) untuk membatasi output ke penyimpanan kunci kustom tertentu.

Untuk penyimpanan kunci khusus, output terdiri dari ID penyimpanan kunci kustom, nama, dan jenis, dan status koneksi penyimpanan kunci. Jika status koneksi`FAILED`, output juga menyertakan a `ConnectionErrorCode` yang menjelaskan alasan kesalahan. Untuk bantuan menafsirkan `ConnectionErrorCode` untuk penyimpanan kunci eksternal, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal.

Untuk penyimpanan kunci eksternal, output juga mencakup `XksProxyConfiguration` elemen. Elemen ini mencakup jenis konektivitas, titik akhir URI proxy, jalur URI proxy, dan ID kunci akses dari kredensi otentikasi proxy.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Misalnya, perintah berikut mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah. Anda dapat menggunakan parameter `Limit` dan `Marker` ke halaman melalui penyimpanan kunci kustom dalam output.

```
$ aws kms describe-custom-key-stores
```

Perintah berikut menggunakan `CustomKeyStoreName` parameter untuk mendapatkan hanya contoh penyimpanan kunci eksternal dengan nama `ExampleXksPublic` ramah. Toko kunci contoh ini menggunakan konektivitas titik akhir publik. Ini terhubung ke proxy penyimpanan kunci eksternal.

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

Perintah berikut mendapatkan contoh penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC. Dalam contoh ini, penyimpanan kunci eksternal terhubung ke proxy penyimpanan kunci eksternal.

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

A `ConnectionState` `Disconnected` menunjukkan bahwa penyimpanan kunci eksternal tidak pernah terhubung atau sengaja terputus dari proxy penyimpanan kunci eksternalnya. Namun, jika upaya untuk menggunakan kunci KMS di penyimpanan kunci eksternal yang terhubung gagal, itu mungkin menunjukkan masalah dengan proxy penyimpanan kunci eksternal atau komponen eksternal lainnya.

Jika `ConnectionState` penyimpanan kunci eksternal adalah `FAILED`, `DescribeCustomKeyStores` responsnya mencakup `ConnectionErrorCode` elemen yang menjelaskan alasan kesalahan.

Misalnya, dalam output berikut, `XKS_PROXY_TIMED_OUT` nilai menunjukkan AWS KMS dapat terhubung ke proxy penyimpanan kunci eksternal, tetapi koneksi gagal karena proxy penyimpanan kunci eksternal tidak merespons AWS KMS dalam waktu yang ditentukan. Jika Anda melihat kode kesalahan koneksi ini berulang kali, beri tahu vendor proxy penyimpanan kunci eksternal Anda. Untuk bantuan dengan hal ini dan kegagalan kesalahan koneksi lainnya, lihat Memecahkan masalah toko kunci eksternal.

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

Memantau toko kunci eksternal

AWS KMSmengumpulkan metrik untuk setiap interaksi dengan toko kunci eksternal dan menerbitkannya di akun Anda. CloudWatch Metrik ini digunakan untuk menghasilkan grafik di bagian pemantauan halaman detail untuk setiap penyimpanan kunci eksternal. Topik berikut merinci cara menggunakan grafik untuk mengidentifikasi dan memecahkan masalah operasional dan konfigurasi yang memengaruhi penyimpanan kunci eksternal Anda. Sebaiknya gunakan CloudWatch metrik untuk menyetel alarm yang memberi tahu Anda saat penyimpanan kunci eksternal Anda tidak berfungsi seperti yang diharapkan. Untuk informasi selengkapnya, lihat Memantau dengan Amazon CloudWatch.

Topik

- Melihat grafik
- Menafsirkan grafik
- Mengatur alarm

Melihat grafik

Anda dapat melihat grafik pada berbagai tingkat detail. Secara default, setiap grafik menggunakan rentang waktu tiga jam dan periode agregasi lima menit. Anda dapat menyesuaikan tampilan grafik di dalam konsol, tetapi perubahan Anda akan kembali ke pengaturan default ketika halaman detail penyimpanan kunci eksternal ditutup atau browser di-refresh. Untuk bantuan terkait CloudWatch terminologi Amazon, lihat CloudWatch Konsep Amazon.

Lihat detail titik data

Data dalam setiap grafik dikumpulkan berdasarkan AWS KMSmetrik. Untuk melihat informasi lebih lanjut tentang titik data tertentu, jeda mouse di atas titik data pada grafik garis. Ini akan menampilkan pop-up dengan informasi lebih lanjut tentang metrik dari mana grafik itu berasal. Setiap item daftar menampilkan nilai dimensi yang direkam pada titik data tersebut. Pop-up menampilkan nilai null (—) jika tidak ada data metrik yang tersedia untuk nilai dimensi pada titik data tersebut. Beberapa grafik merekam beberapa dimensi dan nilai untuk satu titik data. Grafik lain, seperti grafik reliabilitas, menggunakan data yang dikumpulkan oleh metrik untuk menghitung nilai unik. Setiap item daftar dikaitkan dengan warna grafik garis yang berbeda.

Ubah rentang waktu

Untuk mengubah rentang waktu, pilih salah satu rentang waktu yang telah ditentukan di sudut kanan atas bagian pemantauan. Rentang waktu yang telah ditentukan berkisar dari 1 jam hingga 1 minggu (1 jam, 3 jam, 12 jam, 1d, 3d, atau 1w). Ini menyesuaikan rentang waktu untuk semua grafik. Jika Anda ingin melihat satu grafik tertentu dalam rentang waktu yang berbeda, atau jika Anda ingin mengatur rentang waktu khusus, perbesar grafik atau lihat di CloudWatch konsol Amazon.

Memperbesar grafik

Anda dapat menggunakan fitur zoom peta mini untuk fokus pada bagian grafik garis dan bagian grafik yang ditumpuk tanpa mengubah antara tampilan yang diperbesar dan diperbesar. Misalnya, Anda dapat menggunakan fitur zoom peta mini untuk fokus pada puncak dalam grafik, sehingga Anda dapat membandingkan lonjakan dengan grafik lain di bagian pemantauan dari garis waktu yang sama.

1. Pilih dan seret pada area grafik yang ingin Anda fokuskan, dan kemudian lepaskan seretannya.
2. Untuk mengatur ulang zoom, pilih ikon Atur ulang zoom, yang terlihat seperti kaca pembesar dengan simbol minus (-) di dalamnya.

Memperbesar grafik

Untuk memperbesar grafik, pilih ikon menu di sudut kanan atas grafik individual dan pilih Perbesar. Anda juga dapat memilih ikon perbesar yang muncul di sebelah ikon menu saat Anda mengarahkan kursor ke grafik.

Memperbesar grafik memungkinkan Anda untuk memodifikasi tampilan grafik lebih lanjut dengan menentukan periode yang berbeda, rentang waktu khusus, atau interval penyegaran. Perubahan ini akan kembali ke pengaturan default saat Anda menutup tampilan yang diperbesar.

Ubah periode

1. Pilih menu opsi Periode. Secara default, menu ini menampilkan nilai: 5 menit.
2. Pilih periode, periode yang telah ditentukan berkisar dari 1 detik hingga 30 hari.

   Misalnya, Anda dapat memilih tampilan satu menit, yang dapat berguna ketika pemecahan masalah. Atau, pilih tampilan satu jam yang kurang terperinci. Hal ini dapat berguna ketika melihat rentang waktu yang lebih luas (misalnya, 3 hari) sehingga Anda dapat melihat tren dari waktu ke waktu. Untuk informasi selengkapnya, lihat Periode di Panduan CloudWatch Pengguna Amazon.

Ubah rentang waktu atau zona waktu

1. Pilih salah satu rentang waktu yang telah ditentukan, yang berkisar dari 1 jam hingga 1 minggu (1 jam, 3 jam, 12 jam, 1d, 3d, atau 1w). Atau, Anda dapat memilih Kustom untuk mengatur rentang waktu Anda sendiri.
2. Pilih Kustom
   a. Rentang waktu: pilih tab Absolute di sudut kiri atas kotak. Gunakan pemilih kalender atau kotak bidang teks untuk menentukan rentang waktu.
   b. Zona waktu: pilih dropdown di sudut kanan atas kotak. Anda dapat mengubah zona waktu ke UTC atau zona waktu lokal.
3. Setelah Anda menentukan rentang waktu, pilih Terapkan.

Ubah seberapa sering data dalam grafik Anda di-refresh

1. Pilih menu Refresh options di pojok kanan atas.
2. Pilih interval penyegaran (Mati, 10 Detik, 1 Menit, 2 Menit, 5 Menit, atau 15 Menit).

Lihat grafik di konsol Amazon CloudWatch

Grafik di bagian pemantauan berasal dari metrik yang telah ditentukan sebelumnya yang AWS KMS diterbitkan ke Amazon. CloudWatch Anda dapat membukanya di dalam CloudWatch konsol dan menyimpannya ke CloudWatch dasbor. Jika Anda memiliki beberapa toko kunci eksternal, Anda dapat membuka grafik masing-masing CloudWatch dan menyimpannya ke satu dasbor untuk membandingkan kesehatan dan penggunaannya.

Tambahkan ke CloudWatch dasbor

Pilih Tambahkan ke dasbor di sudut kanan atas untuk menambahkan semua grafik ke CloudWatch dasbor Amazon. Anda dapat memilih dasbor yang ada atau membuat yang baru. Untuk informasi tentang penggunaan dasbor ini untuk membuat tampilan grafik dan alarm yang disesuaikan, lihat Menggunakan CloudWatchdasbor Amazon di Panduan Pengguna Amazon CloudWatch .

Lihat dalam CloudWatch metrik

Pilih ikon menu di sudut kanan atas grafik individual dan pilih Lihat dalam metrik untuk melihat grafik ini di CloudWatch konsol Amazon. Dari CloudWatch konsol, Anda dapat menambahkan grafik tunggal ini ke dasbor dan mengubah rentang waktu, periode, dan interval penyegaran. Untuk informasi selengkapnya, lihat, Membuat grafik metrik di CloudWatch Panduan Pengguna Amazon.

Menafsirkan grafik

AWS KMSmenyediakan beberapa grafik untuk memantau kesehatan toko kunci eksternal Anda di dalam AWS KMS konsol. Grafik ini secara otomatis dikonfigurasi dan berasal dari AWS KMSmetrik.

Data grafik dikumpulkan sebagai bagian dari panggilan yang Anda lakukan ke penyimpanan kunci eksternal dan kunci eksternal Anda. Anda mungkin melihat data mengisi grafik selama rentang waktu yang Anda tidak melakukan panggilan apa pun, data ini berasal dari `GetHealthStatus` panggilan berkala yang AWS KMS membuat atas nama Anda untuk memeriksa status proxy penyimpanan kunci eksternal dan pengelola kunci eksternal Anda. Jika grafik Anda menampilkan pesan Tidak ada data yang tersedia, maka tidak ada panggilan yang direkam selama rentang waktu tersebut atau penyimpanan kunci eksternal Anda dalam `DISCONNECTED`keadaan. Anda mungkin dapat

mengidentifikasi waktu penyimpanan kunci eksternal Anda terputus dengan menyesuaikan tampilan Anda ke rentang waktu yang lebih luas.

Topik

- Total permintaan
- Keandalan
- Latensi
- 5 pengecualian teratas
- Hari sertifikat untuk kedaluwarsa

Total permintaan

Jumlah total AWS KMS permintaan yang diterima untuk penyimpanan kunci eksternal tertentu selama rentang waktu tertentu. Gunakan grafik ini untuk menentukan apakah Anda berisiko mengalami pelambatan.

AWS KMSmerekomendasikan bahwa manajer kunci eksternal Anda dapat menangani hingga 1800 permintaan untuk operasi kriptografi per detik. Jika Anda mendekati 540.000 panggilan dalam periode lima menit, Anda berisiko mengalami pelambatan.

Anda dapat memantau jumlah permintaan untuk operasi kriptografi pada kunci KMS di penyimpanan kunci eksternal Anda yang AWS KMS dibatasi dengan metrik. ExternalKeyStoreThrottle

Jika Anda mendapatkan `KMSInvalidStateException` kesalahan yang sangat sering dengan pesan yang menjelaskan bahwa permintaan ditolak “karena tingkat permintaan yang sangat tinggi,” itu mungkin menunjukkan bahwa manajer kunci eksternal atau proxy penyimpanan kunci eksternal Anda tidak dapat mengimbangi tingkat permintaan saat ini. Jika memungkinkan, turunkan tingkat permintaan Anda. Anda juga dapat mempertimbangkan untuk meminta penurunan nilai kuota permintaan toko kunci kustom Anda. Penurunan nilai kuota ini dapat meningkatkan pembatasan, tetapi ini menunjukkan bahwa AWS KMS menolak permintaan berlebih dengan cepat sebelum dikirim ke proxy penyimpanan kunci eksternal atau pengelola kunci eksternal Anda. Untuk meminta pengurangan kuota, silakan kunjungi AWS SupportPusat dan buat kasus.

Grafik permintaan total berasal dari XksProxyErrors metrik, yang mengumpulkan data tentang respons yang berhasil dan tidak berhasil yang AWS KMS diterima dari proxy penyimpanan kunci eksternal Anda. Saat Anda melihat titik data tertentu, pop-up menampilkan nilai `CustomKeyStoreId` dimensi di samping jumlah total AWS KMS permintaan yang direkam pada titik data tersebut. `CustomKeyStoreId`Akan selalu sama.

Keandalan

Persentase AWS KMS permintaan yang proxy penyimpanan kunci eksternal mengembalikan respons yang berhasil atau kesalahan yang tidak dapat dicoba ulang. Gunakan grafik ini untuk mengevaluasi kesehatan operasional proxy penyimpanan kunci eksternal Anda.

Ketika grafik menampilkan nilai kurang dari 100%, ini menunjukkan kasus di mana proxy tidak merespons atau merespons dengan kesalahan yang dapat dicoba ulang. Ini dapat menunjukkan masalah dengan jaringan, lambatnya proxy penyimpanan kunci eksternal atau manajer kunci eksternal, atau bug implementasi.

Jika permintaan menyertakan kredensi buruk dan proxy Anda merespons dengan`AuthenticationFailedException`, grafik akan tetap menunjukkan keandalan 100% karena proxy mengidentifikasi nilai yang salah dalam permintaan API proxy penyimpanan kunci eksternal, dan oleh karena itu kegagalan diharapkan terjadi. Jika persentase grafik reliabilitas Anda 100%, maka proxy penyimpanan kunci eksternal Anda merespons seperti yang diharapkan. Jika grafik menampilkan nilai kurang dari 100%, maka proxy merespons dengan kesalahan yang dapat dicoba ulang atau habis waktu. Misalnya, jika proxy merespons dengan `ThrottlingException` karena tingkat permintaan yang sangat tinggi, itu akan menampilkan persentase keandalan yang lebih rendah karena proxy tidak dapat mengidentifikasi masalah tertentu dalam permintaan yang menyebabkannya gagal. Ini karena kesalahan yang dapat dicoba ulang kemungkinan merupakan masalah sementara yang dapat diselesaikan dengan mencoba kembali permintaan.

Respons kesalahan berikut akan menurunkan persentase keandalan. Anda dapat menggunakan 5 pengecualian teratas grafik dan XksProxyErrors metrik untuk memantau lebih lanjut seberapa sering proxy Anda mengembalikan setiap kesalahan yang dapat dicoba ulang.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

Grafik reliabilitas berasal dari XksProxyErrors metrik, yang mengumpulkan data tentang respons yang berhasil dan tidak berhasil yang AWS KMS diterima dari proxy penyimpanan kunci eksternal Anda. Persentase reliabilitas hanya akan lebih rendah jika respons memiliki `ErrorType` nilai`Retryable`. Saat Anda melihat titik data tertentu, pop-up menampilkan nilai `CustomKeyStoreId` dimensi di samping persentase keandalan untuk AWS KMS permintaan yang direkam pada titik data tersebut. `CustomKeyStoreId`Akan selalu sama.

Sebaiknya gunakan XksProxyErrors metrik untuk membuat CloudWatch alarm yang memberi tahu Anda tentang potensi masalah jaringan dengan memberi tahu Anda ketika lebih dari lima kesalahan yang dapat dicoba ulang direkam dalam periode satu menit. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm Amazon untuk kesalahan yang dapat dicoba ulang.

Latensi

Jumlah milidetik yang diperlukan untuk proxy penyimpanan kunci eksternal untuk menanggapi AWS KMS permintaan. Gunakan grafik ini untuk mengevaluasi kinerja proxy penyimpanan kunci eksternal dan manajer kunci eksternal Anda.

AWS KMSmengharapkan proxy penyimpanan kunci eksternal untuk menanggapi setiap permintaan dalam 250 milidetik. Dalam kasus batas waktu jaringan, AWS KMS akan mencoba lagi permintaan sekali. Jika proxy gagal untuk kedua kalinya, latensi yang direkam adalah batas batas waktu gabungan untuk kedua upaya permintaan dan grafik akan menampilkan sekitar 500 milidetik. Dalam semua kasus lain di mana proxy tidak merespons dalam batas waktu 250 milidetik, latensi yang direkam adalah 250 milidetik. Jika proxy sering habis waktu pada operasi enkripsi dan dekripsi, konsultasikan dengan administrator proxy eksternal Anda. Untuk bantuan memecahkan masalah latensi, lihat. Kesalahan latensi dan batas waktu

Respons lambat mungkin juga menunjukkan bahwa pengelola kunci eksternal Anda tidak dapat menangani lalu lintas permintaan saat ini. AWS KMSmerekomendasikan bahwa manajer kunci eksternal Anda dapat menangani hingga 1800 permintaan untuk operasi kriptografi per detik. Jika pengelola kunci eksternal Anda tidak dapat menangani tarif 1800 permintaan per detik, pertimbangkan untuk meminta penurunan kuota permintaan Anda untuk kunci KMS di toko kunci khusus. Permintaan untuk operasi kriptografi menggunakan kunci KMS di toko kunci eksternal Anda akan gagal dengan cepat dengan pengecualian pelambatan, daripada diproses dan kemudian ditolak oleh proxy penyimpanan kunci eksternal atau manajer kunci eksternal Anda.

Grafik latensi berasal dari XksProxyLatency metrik. Saat Anda melihat titik data tertentu, pop-up menampilkan nilai yang sesuai `KmsOperation` dan `XksOperation` dimensi di samping latensi rata-rata yang direkam untuk operasi pada titik data tersebut. Item daftar diurutkan dari latensi tertinggi ke terendah.

Sebaiknya gunakan XksProxyLatency metrik untuk membuat CloudWatch alarm yang memberi tahu Anda saat latensi Anda mendekati batas waktu tunggu. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm Amazon untuk batas waktu respons.

5 pengecualian teratas

Lima pengecualian teratas untuk operasi kriptografi dan manajemen yang gagal selama rentang waktu tertentu. Gunakan grafik ini untuk melacak kesalahan yang paling sering terjadi, sehingga Anda dapat memprioritaskan upaya teknik Anda.

Jumlah ini mencakup pengecualian yang AWS KMS diterima dari proxy penyimpanan kunci eksternal dan `XksProxyUnreachableException` yang AWS KMS kembali secara internal ketika tidak dapat menjalin komunikasi dengan proxy penyimpanan kunci eksternal.

Tingkat kesalahan yang dapat dicoba ulang yang tinggi mungkin mengindikasikan kesalahan jaringan, sementara tingkat kesalahan yang tidak dapat dicoba ulang yang tinggi mungkin mengindikasikan masalah dengan konfigurasi penyimpanan kunci eksternal Anda. Misalnya, lonjakan `AuthenticationFailedExceptions` menunjukkan perbedaan antara kredensyal otentikasi yang dikonfigurasi AWS KMS dan proxy penyimpanan kunci eksternal. Untuk melihat konfigurasi penyimpanan kunci eksternal Anda, lihatMelihat toko kunci eksternal. Untuk mengedit pengaturan penyimpanan kunci eksternal Anda, lihatMengedit properti penyimpanan kunci eksternal.

Pengecualian yang AWS KMS diterima dari proxy penyimpanan kunci eksternal berbeda dari pengecualian yang AWS KMS dikembalikan ketika operasi gagal. AWS KMSOperasi kriptografi mengembalikan sebuah `KMSInvalidStateException` untuk semua kegagalan yang terkait dengan konfigurasi eksternal atau status koneksi dari penyimpanan kunci eksternal. Untuk mengidentifikasi masalah, gunakan teks pesan kesalahan yang menyertainya.

Tabel berikut menunjukkan pengecualian yang dapat muncul di 5 grafik pengecualian teratas dan pengecualian terkait yang AWS KMS kembali kepada Anda.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Tidak dapat dicoba	**`AccessDeniedException`** Untuk bantuan penyelesaian masalah, lihat Masalah otorisasi proxy.	**`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Tidak dapat dicoba	**`AuthenticationFailedException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan kredensi otentikasi.	**`XksProxyIncorrectAuthenticationCredentialException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Dicoba ulang	**`DependencyTimeoutException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan latensi dan batas waktu.	**`XksProxyUriUnreachableException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Dicoba ulang	**`InternalException`** Proxy penyimpanan kunci eksternal menolak permintaan karena tidak dapat berkomunikasi dengan manajer kunci eksternal. Verifikasi bahwa konfigurasi proxy penyimpanan kunci eksternal sudah benar dan pengelola kunci eksternal tersedia.	**`XksProxyInvalidResponseException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba	**`InvalidCiphertextException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan dekripsi.	**`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba	**`InvalidKeyUsageException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan operasi kriptografi untuk kunci eksternal.	**`XksKeyInvalidConfigurationException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Tidak dapat dicoba	**`InvalidStateException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan operasi kriptografi untuk kunci eksternal.	**`XksKeyInvalidConfigurationException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba	**`InvalidUriPathException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan konfigurasi umum.	**`XksProxyInvalidConfigurationException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba	**`KeyNotFoundException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan kunci eksternal.	**`XksKeyNotFoundException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Dicoba ulang	**`ThrottlingException`** Proxy penyimpanan kunci eksternal menolak permintaan karena tingkat permintaan yang sangat tinggi. Kurangi frekuensi panggilan Anda menggunakan tombol KMS di toko kunci eksternal ini.	**`XksProxyUriUnreachableException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Tidak dapat dicoba	**`UnsupportedOperationException`** Untuk bantuan penyelesaian masalah, lihat Kesalahan operasi kriptografi untuk kunci eksternal.	**`XksKeyInvalidResponseException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Jenis kesalahan	Pengecualian ditampilkan dalam grafik	Pengecualian yang AWS KMS kembali kepada Anda
Tidak dapat dicoba	**`ValidationException`** Untuk bantuan penyelesaian masalah, lihat Masalah proxy.	**`XksProxyInvalidResponseException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.
Dicoba ulang	**`XksProxyUnreachableException`** Jika Anda melihat kesalahan ini berulang kali, verifikasi bahwa proxy penyimpanan kunci eksternal Anda aktif dan terhubung ke jaringan, dan bahwa jalur URI dan titik akhir URI atau nama layanan VPC sudah benar di penyimpanan kunci eksternal Anda.	**`XksProxyUriUnreachableException`** dalam menanggapi `CreateCustomKeyStore` dan `UpdateCustomKeyStore` operasi. **`CustomKeyStoreInvalidStateException`** dalam menanggapi `CreateKey` operasi. **`KMSInvalidStateException`** dalam menanggapi operasi kriptografi.

Grafik 5 pengecualian teratas berasal dari XksProxyErrors metrik. Saat Anda melihat titik data tertentu, pop-up menampilkan nilai `ExceptionName` dimensi di samping berapa kali pengecualian

direkam pada titik data tersebut. Lima item daftar diurutkan dari pengecualian yang paling sering hingga yang paling sedikit.

Sebaiknya gunakan XksProxyErrors metrik untuk membuat CloudWatch alarm yang memberi tahu Anda tentang potensi masalah konfigurasi dengan memberi tahu Anda ketika lebih dari lima kesalahan yang tidak dapat dicoba ulang direkam dalam periode satu menit. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm Amazon untuk kesalahan yang tidak dapat dicoba ulang.

Hari sertifikat untuk kedaluwarsa

Jumlah hari hingga sertifikat TLS untuk titik akhir proxy penyimpanan kunci eksternal Anda (`XksProxyUriEndpoint`) kedaluwarsa. Gunakan grafik ini untuk memantau kedaluwarsa sertifikat TLS Anda yang akan datang.

Ketika sertifikat kedaluwarsa, AWS KMS tidak dapat berkomunikasi dengan proxy penyimpanan kunci eksternal. Semua data yang dilindungi oleh kunci KMS di toko kunci eksternal Anda menjadi tidak dapat diakses sampai Anda memperbarui sertifikat.

Grafik sertifikat hari untuk kedaluwarsa berasal dari XksProxyCertificateDaysToExpire metrik. Kami sangat menyarankan menggunakan metrik ini untuk membuat CloudWatch alarm yang memberi tahu Anda tentang kedaluwarsa yang akan datang. Kedaluwarsa sertifikat dapat mencegah Anda mengakses sumber daya terenkripsi Anda. Atur alarm untuk memberi waktu kepada organisasi Anda untuk memperbarui sertifikat sebelum kedaluwarsa. Untuk informasi selengkapnya, lihat Membuat CloudWatch alarm Amazon untuk kedaluwarsa sertifikat.

Mengatur alarm

Grafik di bagian pemantauan memberikan gambaran umum tentang kesehatan toko kunci eksternal Anda dan kunci KMS di toko kunci eksternal untuk jangka waktu tertentu. Namun, Anda dapat membuat CloudWatch alarm Amazon berdasarkan metrik penyimpanan kunci eksternal untuk memberi tahu Anda bila nilai metrik melebihi ambang batas yang Anda tentukan. Alarm dapat mengirim pesan ke topik Amazon Simple Notification Service (Amazon SNS) atau kebijakan Auto Scaling Amazon EC2. Untuk informasi selengkapnya tentang CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon di CloudWatch Panduan Pengguna Amazon.

Sebelum membuat CloudWatch alarm Amazon, Anda memerlukan topik Amazon SNS. Untuk detailnya, lihat Membuat topik Amazon SNS di CloudWatch Panduan Pengguna Amazon.

Topik

- Membuat CloudWatch alarm Amazon untuk kedaluwarsa sertifikat
- Membuat CloudWatch alarm Amazon untuk batas waktu respons
- Membuat CloudWatch alarm Amazon untuk kesalahan yang dapat dicoba ulang
- Membuat CloudWatch alarm Amazon untuk kesalahan yang tidak dapat dicoba ulang

Membuat CloudWatch alarm Amazon untuk kedaluwarsa sertifikat

Alarm ini menggunakan XksProxyCertificateDaysToExpire metrik yang AWS KMS diterbitkan CloudWatch untuk merekam antisipasi kedaluwarsa sertifikat TLS yang terkait dengan titik akhir proxy penyimpanan kunci eksternal Anda. Anda tidak dapat membuat alarm tunggal untuk semua penyimpanan kunci eksternal di akun Anda atau alarm untuk penyimpanan kunci eksternal yang mungkin Anda buat di masa mendatang.

Sebaiknya atur alarm untuk mengingatkan Anda 10 hari sebelum sertifikat Anda ditetapkan kedaluwarsa, tetapi Anda harus menetapkan ambang batas yang paling sesuai dengan kebutuhan Anda.

Buat alarm

Ikuti petunjuk di Buat CloudWatch alarm berdasarkan ambang statis menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pilih metrik	Pilih KMS, lalu pilih Metrik Sertifikat Proxy XKS. Pilih kotak centang di sebelah `XksProxyCertificateName` yang ingin Anda pantau. Kemudian pilih Pilih metrik.
Statistik	Minimum
Periode	5 menit
Jenis ambang	Statis
Kapanpun...	Kapan pun XksProxyCertificateDaysToExpire`Lower`lebih dari`10`.

Membuat CloudWatch alarm Amazon untuk batas waktu respons

Alarm ini menggunakan XksProxyLatency metrik yang AWS KMS diterbitkan CloudWatch untuk merekam jumlah milidetik yang diperlukan untuk proxy penyimpanan kunci eksternal AWS KMS untuk menanggapi permintaan. Anda tidak dapat membuat alarm tunggal untuk semua penyimpanan kunci eksternal di akun Anda atau alarm untuk penyimpanan kunci eksternal yang mungkin Anda buat di masa mendatang.

AWS KMSmengharapkan proxy penyimpanan kunci eksternal untuk menanggapi setiap permintaan dalam 250 milidetik. Sebaiknya atur alarm untuk mengingatkan Anda ketika proxy penyimpanan kunci eksternal Anda membutuhkan waktu lebih dari 200 milidetik untuk merespons, tetapi Anda harus menetapkan ambang batas yang paling sesuai dengan kebutuhan Anda.

Buat alarm

Ikuti petunjuk di Buat CloudWatch alarm berdasarkan ambang statis menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pilih metrik	Pilih KMS, lalu pilih Metrik Latensi Proxy XKS. Pilih kotak centang di sebelah `KmsOperation` yang ingin Anda pantau. Kemudian pilih Pilih metrik.
Statistik	Rata-rata
Periode	5 menit
Jenis ambang	Statis
Kapanpun...	Kapan pun XksProxyLatency`Greater`lebih dari`200`.

Membuat CloudWatch alarm Amazon untuk kesalahan yang dapat dicoba ulang

Alarm ini menggunakan XksProxyErrors metrik yang AWS KMS diterbitkan CloudWatch untuk mencatat jumlah pengecualian yang terkait dengan AWS KMS permintaan ke proxy penyimpanan kunci eksternal Anda. Anda tidak dapat membuat alarm tunggal untuk semua penyimpanan kunci

eksternal di akun Anda atau alarm untuk penyimpanan kunci eksternal yang mungkin Anda buat di masa mendatang.

Kesalahan yang dapat dicoba ulang akan menurunkan persentase keandalan Anda dan dapat menunjukkan kesalahan jaringan. Kami merekomendasikan pengaturan alarm untuk mengingatkan Anda ketika lebih dari lima kesalahan yang dapat dicoba ulang dicatat dalam periode satu menit, tetapi Anda harus menetapkan ambang batas yang paling sesuai dengan kebutuhan Anda.

Ikuti petunjuk di Buat CloudWatch alarm berdasarkan ambang statis menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pilih metrik	Pilih tab Kueri. Pilih `AWS/KMS` untuk Namespace. Masukkan `SUM(XksProxyErrors)` untuk nama Metrik. Masukkan `ErrorType = Retryable` untuk Filter oleh. Pilih Jalankan. Kemudian pilih Pilih metrik.
Label	*Kesalahan yang dapat dicoba ulang*
Periode	1 menit
Jenis ambang	Statis
Kapanpun...	Setiap kali q1 adalah `Greater` dari5.

Membuat CloudWatch alarm Amazon untuk kesalahan yang tidak dapat dicoba ulang

Alarm ini menggunakan XksProxyErrors metrik yang AWS KMS diterbitkan CloudWatch untuk mencatat jumlah pengecualian yang terkait dengan AWS KMS permintaan ke proxy penyimpanan kunci eksternal Anda. Anda tidak dapat membuat alarm tunggal untuk semua penyimpanan kunci eksternal di akun Anda atau alarm untuk penyimpanan kunci eksternal yang mungkin Anda buat di masa mendatang.

Kesalahan yang tidak dapat dicoba ulang dapat menunjukkan masalah dengan konfigurasi penyimpanan kunci eksternal Anda. Kami merekomendasikan pengaturan alarm untuk mengingatkan Anda ketika lebih dari lima kesalahan yang tidak dapat dicoba ulang dicatat dalam periode satu menit, tetapi Anda harus menetapkan ambang batas yang paling sesuai dengan kebutuhan Anda.

Ikuti petunjuk di Buat CloudWatch alarm berdasarkan ambang statis menggunakan nilai yang diperlukan berikut. Untuk bidang lain, terima nilai default dan berikan nama seperti yang diminta.

Bidang	Nilai
Pilih metrik	Pilih tab Kueri. Pilih `AWS/KMS` untuk Namespace. Masukkan `SUM(XksProxyErrors)` untuk nama Metrik. Masukkan `ErrorType = Non-retryable` untuk Filter oleh. Pilih Jalankan. Kemudian pilih Pilih metrik.
Label	*Kesalahan yang tidak dapat dicoba ulang*
Periode	1 menit
Jenis ambang	Statis
Kapanpun...	Setiap kali q1 adalah `Greater` dari5.

Menghubungkan dan memutuskan penyimpanan kunci eksternal

Toko kunci eksternal baru tidak terhubung. Untuk membuat dan menggunakan AWS KMS keys di toko kunci eksternal Anda, Anda perlu menghubungkan toko kunci eksternal Anda ke proxy penyimpanan kunci eksternal. Anda dapat menghubungkan dan memutuskan penyimpanan kunci eksternal Anda kapan saja, dan melihat status koneksinya.

Sementara penyimpanan kunci eksternal Anda terputus, AWS KMS tidak dapat berkomunikasi dengan proxy penyimpanan kunci eksternal Anda. Akibatnya, Anda dapat melihat dan mengelola toko kunci eksternal Anda dan kunci KMS yang ada. Namun, Anda tidak dapat membuat kunci KMS di toko kunci eksternal Anda, atau menggunakan kunci KMS-nya dalam operasi kriptografi. Anda mungkin perlu memutuskan penyimpanan kunci eksternal Anda di beberapa titik, seperti saat

mengedit propertinya, tetapi rencanakan dengan tepat. Memutuskan sambungan toko kunci dapat mengganggu pengoperasian AWS layanan yang menggunakan kunci KMS-nya.

Anda tidak diharuskan untuk menghubungkan toko kunci eksternal Anda. Anda dapat meninggalkan penyimpanan kunci eksternal dalam keadaan terputus tanpa batas waktu dan menghubungkannya hanya ketika Anda perlu menggunakannya. Namun, Anda mungkin ingin menguji koneksi secara berkala untuk memverifikasi bahwa pengaturan sudah benar dan dapat tersambung.

Saat Anda memutuskan penyimpanan kunci khusus, kunci KMS di toko kunci menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhiLayanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Untuk detailnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

> **Note**
>
> Penyimpanan kunci eksternal berada dalam `DISCONNECTED` keadaan hanya ketika toko kunci tidak pernah terhubung atau Anda secara eksplisit memutuskannya. `CONNECTED`Status tidak menunjukkan bahwa penyimpanan kunci eksternal atau komponen pendukungnya beroperasi secara efisien. Untuk informasi tentang kinerja komponen penyimpanan kunci eksternal Anda, lihat grafik di bagian Monitoring pada halaman detail untuk setiap penyimpanan kunci eksternal. Untuk detailnya, lihat Memantau toko kunci eksternal. Manajer kunci eksternal Anda mungkin menyediakan metode tambahan untuk menghentikan dan memulai kembali komunikasi antara penyimpanan kunci AWS KMS eksternal dan proxy penyimpanan kunci eksternal Anda, atau antara proxy penyimpanan kunci eksternal dan manajer kunci eksternal. Untuk detailnya, lihat dokumentasi pengelola kunci eksternal Anda.

Topik

- Menghubungkan toko kunci eksternal
- Memutuskan sambungan penyimpanan kunci eksternal
- Status koneksi
- Hubungkan toko kunci eksternal (konsol)
- Hubungkan penyimpanan kunci eksternal (API)
- Putuskan sambungan penyimpanan kunci eksternal (konsol)

- Putuskan sambungan penyimpanan kunci eksternal (API)

Menghubungkan toko kunci eksternal

Ketika penyimpanan kunci eksternal Anda terhubung ke proxy penyimpanan kunci eksternal, Anda dapat membuat kunci KMS di toko kunci eksternal Anda dan menggunakan kunci KMS yang ada dalam operasi kriptografi.

Proses yang menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal berbeda berdasarkan konektivitas penyimpanan kunci eksternal.

- Saat Anda menghubungkan penyimpanan kunci eksternal dengan konektivitas titik akhir publik, AWS KMS mengirimkan GetHealthStatus permintaan ke proxy penyimpanan kunci eksternal untuk memvalidasi titik akhir URI proxy, jalur URI proxy, dan kredensi otentikasi proxy. Respons yang berhasil dari proxy mengonfirmasi bahwa titik akhir URI proxy dan jalur URI proxy akurat dan dapat diakses, dan bahwa proxy mengautentikasi permintaan yang ditandatangani dengan kredensi otentikasi proxy untuk penyimpanan kunci eksternal.
- Saat Anda menghubungkan penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC ke proxy penyimpanan kunci eksternal, AWS KMS lakukan hal berikut:
  - Mengonfirmasi bahwa domain untuk nama DNS pribadi yang ditentukan dalam titik akhir URI proxy telah diverifikasi.
  - Membuat titik akhir antarmuka dari AWS KMS VPC ke layanan titik akhir VPC Anda.
  - Membuat zona host pribadi untuk nama DNS pribadi yang ditentukan dalam titik akhir URI proxy
  - Mengirim GetHealthStatuspermintaan ke proxy penyimpanan kunci eksternal. Respons yang berhasil dari proxy mengonfirmasi bahwa titik akhir URI proxy dan jalur URI proxy akurat dan dapat diakses, dan bahwa proxy mengautentikasi permintaan yang ditandatangani dengan kredensi otentikasi proxy untuk penyimpanan kunci eksternal.

Operasi connect memulai proses menghubungkan toko kunci kustom Anda, tetapi menghubungkan kunci eksternal menyimpannya proxy eksternal membutuhkan waktu sekitar lima menit. Respons sukses dari operasi koneksi tidak menunjukkan bahwa penyimpanan kunci eksternal terhubung. Untuk mengonfirmasi bahwa koneksi berhasil, gunakan AWS KMS konsol atau DescribeCustomKeyStoresoperasi untuk melihat status koneksi eksternal penyimpanan kunci Anda.

Ketika status koneksi`FAILED`, kode kesalahan koneksi ditampilkan di AWS KMS konsol dan ditambahkan ke `DescribeCustomKeyStore` respons. Untuk bantuan menafsirkan kode kesalahan koneksi, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal.

Memutuskan sambungan penyimpanan kunci eksternal

Ketika Anda memutuskan penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC dari proxy penyimpanan kunci eksternalAWS KMS, menghapus titik akhir antarmuka ke layanan titik akhir VPC dan menghapus infrastruktur jaringan yang dibuat untuk mendukung koneksi. Tidak diperlukan proses yang setara untuk penyimpanan kunci eksternal dengan konektivitas titik akhir publik. Tindakan ini tidak memengaruhi layanan titik akhir VPC atau komponen pendukungnya, dan tidak memengaruhi proxy penyimpanan kunci eksternal atau komponen eksternal apa pun.

Sementara penyimpanan kunci eksternal terputus, AWS KMS tidak mengirim permintaan apa pun ke proxy penyimpanan kunci eksternal. Status koneksi dari toko kunci eksternal adalah`DISCONNECTED`. Kunci KMS di penyimpanan kunci eksternal yang terputus berada dalam keadaan `UNAVAILABLE` kunci (kecuali jika sedang menunggu penghapusan), yang berarti bahwa mereka tidak dapat digunakan dalam operasi kriptografi. Namun, Anda masih dapat melihat dan mengelola toko kunci eksternal Anda dan kunci KMS yang ada.

Keadaan terputus dirancang untuk bersifat sementara dan reversibel. Anda dapat menghubungkan kembali toko kunci eksternal Anda kapan saja. Biasanya, tidak diperlukan konfigurasi ulang. Namun, jika ada properti dari proxy penyimpanan kunci eksternal yang terkait telah berubah saat terputus, seperti rotasi kredensi otentikasi proksi, Anda harus mengedit pengaturan penyimpanan kunci eksternal sebelum menyambung kembali.

> **Note**
>
> Sementara toko kunci khusus terputus, semua upaya untuk membuat kunci KMS di toko kunci khusus atau menggunakan kunci KMS yang ada dalam operasi kriptografi akan gagal. Tindakan ini dapat mencegah pengguna menyimpan dan mengakses data sensitif.

Untuk memperkirakan efek pemutusan penyimpanan kunci eksternal Anda dengan lebih baik, identifikasi kunci KMS di penyimpanan kunci eksternal dan tentukan penggunaannya di masa lalu.

Anda dapat memutuskan sambungan penyimpanan kunci eksternal karena alasan seperti berikut:

- Untuk mengedit propertinya. Anda dapat mengedit nama penyimpanan kunci kustom, jalur URI proxy, dan kredensi otentikasi proxy saat penyimpanan kunci eksternal terhubung. Namun, untuk mengedit jenis konektivitas proxy, titik akhir URI proxy, atau nama layanan titik akhir VPC, Anda harus terlebih dahulu memutuskan penyimpanan kunci eksternal. Untuk detailnya, lihat Mengedit properti penyimpanan kunci eksternal.

- Untuk menghentikan semua komunikasi antara AWS KMS dan proxy penyimpanan kunci eksternal. Anda juga dapat menghentikan komunikasi antara AWS KMS dan proxy Anda dengan menonaktifkan layanan endpoint atau VPC endpoint Anda. Selain itu, proxy penyimpanan kunci eksternal atau perangkat lunak manajemen kunci Anda mungkin menyediakan mekanisme tambahan untuk AWS KMS mencegah komunikasi dengan proxy atau untuk mencegah proxy mengakses manajer kunci eksternal Anda.
- Untuk menonaktifkan semua kunci KMS di toko kunci eksternal. Anda dapat menonaktifkan dan mengaktifkan kembali kunci KMS di toko kunci eksternal dengan menggunakan AWS KMS konsol atau operasi. DisableKey Operasi ini selesai dengan cepat (tergantung pada konsistensi akhirnya), tetapi mereka bertindak pada satu kunci KMS pada satu waktu. Memutuskan sambungan penyimpanan kunci eksternal mengubah status kunci dari semua kunci KMS di penyimpanan kunci eksternal`Unavailable`, yang mencegahnya digunakan dalam operasi kriptografi apa pun.
- Untuk memperbaiki upaya koneksi yang gagal. Jika upaya untuk menghubungkan penyimpanan kunci eksternal gagal (status koneksi penyimpanan kunci kustom`FAILED`), Anda harus memutuskan penyimpanan kunci eksternal sebelum Anda mencoba menghubungkannya lagi.

Status koneksi

Menghubungkan dan memutuskan sambungan mengubah status koneksi toko kunci kustom Anda. Nilai status koneksi sama untuk penyimpanan AWS CloudHSM kunci dan penyimpanan kunci eksternal.

Untuk melihat status koneksi penyimpanan kunci kustom Anda, gunakan DescribeCustomKeyStoresoperasi atau AWS KMS konsol. Status koneksi muncul di setiap tabel penyimpanan kunci kustom, di bagian konfigurasi umum dari halaman detail untuk setiap toko kunci kustom, dan pada tab konfigurasi kriptografi kunci KMS di toko kunci khusus. Untuk detailnya, lihat Melihat toko AWS CloudHSM kunci dan Melihat toko kunci eksternal.

Toko kunci khusus dapat memiliki salah satu status koneksi berikut:

- `CONNECTED`: Toko kunci khusus terhubung ke toko kunci pendukungnya. Anda dapat membuat dan menggunakan kunci KMS di toko kunci khusus.

  Toko kunci pendukung untuk toko AWS CloudHSM kunci adalah AWS CloudHSM cluster terkaitnya. Penyimpanan kunci pendukung untuk penyimpanan kunci eksternal adalah proxy penyimpanan kunci eksternal dan manajer kunci eksternal yang didukungnya.

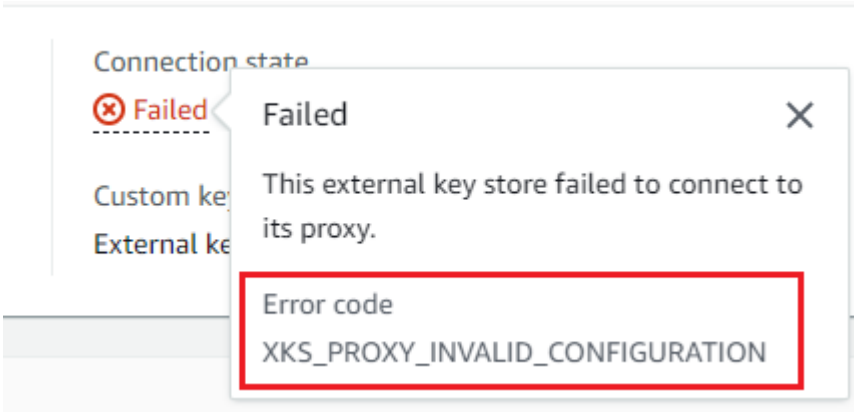
Status CONNECTED berarti bahwa koneksi berhasil dan penyimpanan kunci khusus belum sengaja terputus. Itu tidak menunjukkan bahwa koneksi beroperasi dengan benar. Untuk informasi tentang status AWS CloudHSM klaster yang terkait dengan penyimpanan AWS CloudHSM kunci Anda, lihat Mendapatkan CloudWatch metrik AWS CloudHSM di Panduan AWS CloudHSM Pengguna. Untuk informasi tentang status dan pengoperasian penyimpanan kunci eksternal Anda, lihat grafik di bagian Pemantauan halaman detail untuk setiap penyimpanan kunci eksternal. Untuk detailnya, lihat Memantau toko kunci eksternal.

- `CONNECTING`: Proses menghubungkan toko kunci khusus sedang berlangsung. Ini adalah keadaan sementara.
- `DISCONNECTED`: Toko kunci khusus tidak pernah terhubung ke dukungannya, atau sengaja terputus dengan menggunakan AWS KMS konsol atau operasi. DisconnectCustomKeyStore
- `DISCONNECTING`: Proses pemutusan penyimpanan kunci khusus sedang berlangsung. Ini adalah keadaan sementara.
- `FAILED`: Upaya untuk menghubungkan toko kunci kustom gagal. `ConnectionErrorCode`Dalam DescribeCustomKeyStoresrespon menunjukkan masalah.

Untuk menghubungkan toko kunci khusus, status koneksinya harus`DISCONNECTED`. Jika status koneksi`FAILED`, gunakan `ConnectionErrorCode` untuk mengidentifikasi dan menyelesaikan masalah. Kemudian lepaskan penyimpanan kunci khusus sebelum mencoba menghubungkannya lagi. Untuk bantuan dengan kegagalan koneksi, lihat Kesalahan koneksi penyimpanan kunci eksternal. Untuk bantuan menanggapi kode kesalahan koneksi, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal.

Untuk melihat kode kesalahan koneksi:

- Dalam DescribeCustomKeyStorestanggapannya, lihat nilai `ConnectionErrorCode` elemen. Elemen ini muncul dalam `DescribeCustomKeyStores` respons hanya ketika `ConnectionState` ada`FAILED`.
- Untuk melihat kode kesalahan koneksi di AWS KMS konsol, pada halaman detail untuk penyimpanan kunci eksternal dan arahkan kursor ke nilai Gagal.



Hubungkan toko kunci eksternal (konsol)

Anda dapat menggunakan AWS KMS konsol untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Pilih baris toko kunci eksternal yang ingin Anda sambungkan.

   Jika status koneksi penyimpanan kunci eksternal GAGAL, Anda harus memutuskan penyimpanan kunci eksternal sebelum Anda menghubungkannya.
5. Dari menu Key Store Actions, pilih Connect.

Proses koneksi biasanya memakan waktu sekitar lima menit untuk diselesaikan. Ketika operasi selesai, status koneksi berubah menjadi CONNECTED.

Jika status koneksi Gagal, arahkan kursor ke status koneksi untuk melihat kode kesalahan koneksi, yang menjelaskan penyebab kesalahan. Untuk bantuan menanggapi kode kesalahan koneksi, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal. Untuk menghubungkan penyimpanan kunci eksternal dengan status koneksi Gagal, Anda harus terlebih dahulu memutuskan penyimpanan kunci kustom.

Hubungkan penyimpanan kunci eksternal (API)

Untuk menghubungkan toko kunci eksternal yang terputus, gunakan ConnectCustomKeyStoreoperasi.

Sebelum menghubungkan, status koneksi penyimpanan kunci eksternal harusDISCONNECTED. Jika keadaan koneksi saat iniFAILED, lepaskan penyimpanan kunci eksternal, lalu sambungkan.

Proses koneksi memakan waktu sekitar lima menit untuk menyelesaikannya. Kecuali gagal dengan cepat, ConnectCustomKeyStore mengembalikan respons HTTP 200 dan objek JSON tanpa properti. Namun, respons awal ini tidak menunjukkan bahwa koneksi berhasil. Untuk menentukan apakah penyimpanan kunci eksternal terhubung, lihat status koneksi dalam DescribeCustomKeyStoresrespons.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Untuk mengidentifikasi penyimpanan kunci eksternal, gunakan ID toko kunci kustom. Anda dapat menemukan ID di halaman Toko kunci kustom di konsol atau dengan menggunakan DescribeCustomKeyStoresoperasi. Sebelum menjalankan contoh ini, ganti contoh ID dengan yang valid.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

ConnectCustomKeyStoreOperasi tidak mengembalikan ConnectionState responsnya. Untuk memverifikasi bahwa penyimpanan kunci eksternal terhubung, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah Anda. Namun Anda dapat menggunakan parameter CustomKeyStoreId atau CustomKeyStoreName (tetapi tidak keduanya) untuk membatasi respons ke penyimpanan kunci kustom tertentu. ConnectionStateNilai CONNECTED menunjukkan bahwa penyimpanan kunci eksternal terhubung ke proxy penyimpanan kunci eksternal.

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

Jika `ConnectionState` nilai dalam `DescribeCustomKeyStores` respons adalah`FAILED`, `ConnectionErrorCode` elemen menunjukkan alasan kegagalan.

Dalam contoh berikut, `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` nilai untuk `ConnectionErrorCode` menunjukkan bahwa tidak AWS KMS dapat menemukan layanan titik akhir VPC yang digunakannya untuk berkomunikasi dengan proxy penyimpanan kunci eksternal. Pastikan `XksProxyVpcEndpointServiceName` sudah benar, prinsipal AWS KMS layanan adalah prinsipal yang diizinkan pada layanan titik akhir VPC Amazon, dan bahwa layanan titik akhir VPC tidak memerlukan penerimaan permintaan koneksi. Untuk bantuan menanggapi kode kesalahan koneksi, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal.

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

Putuskan sambungan penyimpanan kunci eksternal (konsol)

Anda dapat menggunakan AWS KMS konsol untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal. Proses ini memakan waktu sekitar 5 menit untuk menyelesaikannya.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Pilih baris toko kunci eksternal yang ingin Anda putuskan.
5. Dari menu Key Store Actions, pilih Disconnect.

Saat operasi selesai, status koneksi berubah dari MENGHUBUNGKAN menjadi TERPUTUS. Jika operasi gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Kesalahan koneksi penyimpanan kunci eksternal.

Putuskan sambungan penyimpanan kunci eksternal (API)

Untuk memutuskan sambungan penyimpanan kunci eksternal yang terhubung, gunakan DisconnectCustomKeyStoreoperasi. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti. Prosesnya memakan waktu sekitar lima menit untuk menyelesaikannya. Untuk menemukan status koneksi penyimpanan kunci eksternal, gunakan DescribeCustomKeyStoresoperasi.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Contoh ini memutus penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC. Sebelum menjalankan contoh ini, ganti contoh ID penyimpanan kunci kustom dengan yang valid.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Untuk memverifikasi bahwa penyimpanan kunci eksternal terputus, gunakan DescribeCustomKeyStoresoperasi. Secara default, operasi ini mengembalikan semua penyimpanan kunci kustom di akun dan Wilayah Anda. Namun Anda dapat menggunakan parameter

`CustomKeyStoreId` dan `CustomKeyStoreName` (tetapi tidak keduanya) untuk membatasi respons ke penyimpanan kunci kustom tertentu. `ConnectionState`Nilai `DISCONNECTED` menunjukkan bahwa contoh penyimpanan kunci eksternal ini tidak lagi terhubung ke proxy penyimpanan kunci eksternal.

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

Menghapus penyimpanan kunci eksternal

Ketika Anda menghapus penyimpanan kunci eksternal, AWS KMS menghapus semua metadata tentang penyimpanan kunci eksternal dariAWS KMS, termasuk informasi tentang proxy penyimpanan kunci eksternal. Operasi ini tidak memengaruhi proxy penyimpanan kunci eksternal, pengelola kunci eksternal, kunci eksternal, atau AWS sumber daya apa pun yang Anda buat untuk mendukung penyimpanan kunci eksternal, seperti VPC Amazon atau layanan titik akhir VPC.

Sebelum Anda menghapus penyimpanan kunci eksternal, Anda harus menghapus semua kunci KMS dari toko kunci dan memutuskan penyimpanan kunci dari proxy penyimpanan kunci eksternal. Jika tidak, upaya untuk menghapus penyimpanan kunci gagal.

Menghapus penyimpanan kunci eksternal tidak dapat diubah, tetapi Anda dapat membuat penyimpanan kunci eksternal baru dan mengaitkannya dengan proxy penyimpanan kunci eksternal dan manajer kunci eksternal yang sama. Namun, Anda tidak dapat membuat ulang kunci KMS enkripsi simetris di penyimpanan kunci eksternal, bahkan Anda memiliki akses ke materi kunci

eksternal yang sama. AWS KMStermasuk metadata dalam ciphertext simetris yang unik untuk setiap kunci KMS. Fitur keamanan ini memastikan bahwa hanya kunci KMS yang mengenkripsi data yang dapat mendekripsi itu.

Alih-alih menghapus toko kunci eksternal, pertimbangkan untuk memutusnya. Sementara toko kunci eksternal terputus, Anda dapat mengelola penyimpanan kunci eksternal dan itu AWS KMS keys tetapi Anda tidak dapat membuat atau menggunakan kunci KMS di toko kunci eksternal. Anda dapat menghubungkan kembali penyimpanan kunci eksternal kapan saja dan melanjutkan menggunakan kunci KMS-nya untuk mengenkripsi dan mendekripsi data. Tidak ada biaya untuk proxy penyimpanan kunci eksternal yang terputus atau kunci KMS yang tidak tersedia.

Topik

- Hapus toko kunci eksternal (konsol)
- Hapus penyimpanan kunci eksternal (API)

Hapus toko kunci eksternal (konsol)

Anda dapat menggunakan AWS KMS konsol untuk menghapus toko kunci eksternal.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Temukan baris yang mewakili penyimpanan kunci eksternal yang ingin Anda hapus. Jika status koneksi penyimpanan kunci eksternal tidak TERPUTUS, Anda harus memutuskan penyimpanan kunci eksternal sebelum Anda menghapusnya.
5. Dari menu Key Store Actions, pilih Delete.

Ketika operasi selesai, pesan sukses muncul dan penyimpanan kunci eksternal tidak lagi muncul di daftar toko kunci. Jika operasi gagal, muncul pesan kesalahan yang menjelaskan masalah dan memberikan bantuan tentang cara memperbaikinya. Jika Anda memerlukan bantuan lebih lanjut, lihat Memecahkan masalah toko kunci eksternal.

Hapus penyimpanan kunci eksternal (API)

Untuk menghapus penyimpanan kunci eksternal, gunakan DeleteCustomKeyStoreoperasi. Jika operasi berhasil, AWS KMS mengembalikan respons HTTP 200 dan objek JSON tanpa properti.

Untuk memulai, lepaskan penyimpanan kunci eksternal. Sebelum menjalankan perintah ini, ganti contoh ID penyimpanan kunci kustom dengan yang valid.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Setelah penyimpanan kunci eksternal terputus, Anda dapat menggunakan DeleteCustomKeyStoreoperasi untuk menghapusnya.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Untuk mengonfirmasi bahwa toko kunci eksternal dihapus, gunakan DescribeCustomKeyStoresoperasi.

```
$ aws kms describe-custom-key-stores

{
    "CustomKeyStores": []
}
```

Jika Anda menentukan nama toko kunci kustom atau ID yang tidak ada lagi, AWS KMS mengembalikan `CustomKeyStoreNotFoundException` pengecualian.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0

An error occurred (CustomKeyStoreNotFoundException) when calling the
 DescribeCustomKeyStore operation:
```

Mengelola kunci KMS di toko kunci eksternal

Untuk membuat, melihat, mengelola, menggunakan, dan menjadwalkan penghapusan kunci KMS di toko kunci eksternal, Anda menggunakan prosedur yang sangat mirip dengan yang Anda gunakan untuk kunci KMS lainnya. Namun, ketika Anda membuat kunci KMS di penyimpanan kunci eksternal, Anda menentukan penyimpanan kunci eksternal dan kunci eksternal. Saat Anda menggunakan kunci KMS di penyimpanan kunci eksternal, operasi enkripsi dan dekripsi dilakukan oleh manajer kunci eksternal Anda menggunakan kunci eksternal yang ditentukan.

AWS KMStidak dapat membuat, melihat, memperbarui, atau menghapus kunci kriptografi apa pun di pengelola kunci eksternal Anda. AWS KMStidak pernah langsung mengakses manajer kunci

eksternal Anda atau kunci eksternal apa pun. Semua permintaan untuk operasi kriptografi dimediasi oleh proxy penyimpanan kunci eksternal Anda. Untuk menggunakan kunci KMS di penyimpanan kunci eksternal, penyimpanan kunci eksternal yang meng-host kunci KMS harus terhubung ke proxy penyimpanan kunci eksternal.

Fitur yang didukung

Selain prosedur yang dibahas di bagian ini, Anda dapat melakukan hal berikut dengan kunci KMS di toko kunci eksternal:

- Gunakan kebijakan utama, kebijakan IAM, dan hibah untuk mengontrol akses ke kunci KMS.
- Aktifkan dan nonaktifkan tombol KMS. Tindakan ini tidak memengaruhi kunci eksternal di manajer kunci eksternal Anda.
- Tetapkan tag dan buat alias, dan gunakan kontrol akses berbasis atribut (ABAC) untuk mengotorisasi akses ke kunci KMS.
- Gunakan kunci KMS Layanan AWSyang terintegrasi dengan AWS KMS dan dukung kunci yang dikelola pelanggan.

Fitur yang tidak didukung

- Toko kunci eksternal hanya mendukung kunci KMS enkripsi simetris. Anda tidak dapat membuat kunci KMS HMAC atau kunci KMS asimetris di penyimpanan kunci eksternal.
- GenerateDataKeyPairdan tidak GenerateDataKeyPairWithoutPlaintextdidukung pada kunci KMS di toko kunci eksternal.
- Anda tidak dapat menggunakan AWS CloudFormationtemplate untuk membuat penyimpanan kunci eksternal atau kunci KMS di penyimpanan kunci eksternal.
- Tombol Multi-Region tidak didukung di penyimpanan kunci eksternal.
- Kunci KMS dengan bahan kunci impor tidak didukung di toko kunci eksternal.
- Rotasi tombol otomatis tidak didukung untuk kunci KMS di penyimpanan kunci eksternal.

Topik

- Membuat kunci KMS di toko kunci eksternal
- Melihat tombol KMS di toko kunci eksternal
- Menggunakan kunci KMS di toko kunci eksternal
- Penjadwalan penghapusan kunci KMS dari toko kunci eksternal

Membuat kunci KMS di toko kunci eksternal

Setelah Anda membuat dan menghubungkan toko kunci eksternal Anda, Anda dapat membuat AWS KMS keysdi toko kunci Anda. Mereka harus enkripsi simetris kunci KMS dengan nilai asal dari External key store ()`EXTERNAL_KEY_STORE`. Anda tidak dapat membuat kunci KMS asimetris, kunci KMS HMAC atau kunci KMS dengan bahan kunci yang diimpor di toko kunci khusus. Selain itu, Anda tidak dapat menggunakan kunci KMS enkripsi simetris di toko kunci khusus untuk menghasilkan pasangan kunci data asimetris.

Kunci KMS di toko kunci eksternal mungkin memiliki latensi, daya tahan, dan ketersediaan yang lebih buruk daripada kunci KMS standar karena tergantung pada komponen yang berada di luar. AWS Sebelum membuat atau menggunakan kunci KMS di toko kunci eksternal, verifikasi bahwa Anda memerlukan kunci dengan properti penyimpanan kunci eksternal.

> **Note**
>
> Beberapa manajer kunci eksternal menyediakan metode yang lebih sederhana untuk membuat kunci KMS di toko kunci eksternal. Untuk detailnya, lihat dokumentasi pengelola kunci eksternal Anda.

Untuk membuat kunci KMS di toko kunci eksternal Anda, Anda menentukan yang berikut ini:

- ID toko kunci eksternal Anda.
- Asal bahan utama dari toko kunci Eksternal (`EXTERNAL_KEY_STORE`).
- ID kunci eksternal yang ada di pengelola kunci eksternal yang terkait dengan penyimpanan kunci eksternal Anda. Kunci eksternal ini berfungsi sebagai bahan utama untuk kunci KMS. Anda tidak dapat mengubah ID kunci eksternal setelah Anda membuat kunci KMS.

  AWS KMSmenyediakan ID kunci eksternal ke proxy penyimpanan kunci eksternal Anda dalam permintaan untuk operasi enkripsi dan dekripsi. AWS KMStidak dapat langsung mengakses manajer kunci eksternal Anda atau kunci kriptografinya.

Selain kunci eksternal, kunci KMS di toko kunci eksternal juga memiliki bahan AWS KMS utama. Semua data yang dienkripsi di bawah kunci KMS pertama kali dienkripsi AWS KMS menggunakan bahan kunci kunci dan kemudian oleh manajer AWS KMS kunci eksternal Anda menggunakan kunci eksternal Anda. Proses enkripsi ganda ini memastikan bahwa ciphertext yang dilindungi oleh kunci

KMS di penyimpanan kunci eksternal setidaknya sekuat ciphertext yang hanya dilindungi oleh. AWS KMS Untuk detailnya, lihat Cara kerja toko kunci eksternal.

Ketika `CreateKey` operasi berhasil, status kunci dari kunci KMS baru adalah. `Enabled` Ketika Anda melihat kunci KMS di penyimpanan kunci eksternal, Anda dapat melihat properti tipikal, seperti ID kunci, spesifikasi kunci, penggunaan kunci, status kunci, dan tanggal pembuatan. Tetapi Anda juga dapat melihat ID dan status koneksi dari penyimpanan kunci eksternal dan ID kunci eksternal.

Jika upaya Anda untuk membuat kunci KMS di penyimpanan kunci eksternal gagal, gunakan pesan kesalahan untuk mengidentifikasi penyebabnya. Ini mungkin menunjukkan bahwa penyimpanan kunci eksternal tidak terhubung (`CustomKeyStoreInvalidStateException`), bahwa proxy penyimpanan kunci eksternal Anda tidak dapat menemukan kunci eksternal dengan ID kunci eksternal yang ditentukan (`XksKeyNotFoundException`), atau bahwa kunci eksternal sudah dikaitkan dengan kunci KMS di penyimpanan `XksKeyAlreadyInUseException` kunci eksternal yang sama.

.

Untuk contoh AWS CloudTrail log operasi yang membuat kunci KMS di penyimpanan kunci eksternal, lihatCreateKey.

Topik

- Persyaratan untuk kunci KMS di toko kunci eksternal
- Buat kunci KMS di toko kunci eksternal (konsol)
- Buat kunci KMS di penyimpanan kunci eksternal (AWS KMSAPI)

Persyaratan untuk kunci KMS di toko kunci eksternal

Untuk membuat kunci KMS di penyimpanan kunci eksternal, properti berikut diperlukan dari penyimpanan kunci eksternal, kunci KMS, dan kunci eksternal yang berfungsi sebagai bahan kunci kriptografi eksternal untuk kunci KMS.

Persyaratan toko kunci eksternal

- Harus terhubung ke proxy penyimpanan kunci eksternal.

  Untuk melihat status koneksi penyimpanan kunci eksternal Anda, lihatMelihat toko kunci eksternal. Untuk menghubungkan toko kunci eksternal Anda, lihatMenghubungkan dan memutuskan penyimpanan kunci eksternal.

Persyaratan utama KMS

Anda tidak dapat mengubah properti ini setelah Anda membuat kunci KMS.

- Spesifikasi kunci: SYMMETRIC_DEFAULT
- Penggunaan kunci: ENCRYPT_DECRYPT
- Asal bahan utama: EXTERNAL_KEY_STORE
- Multi-Wilayah: SALAH

Persyaratan kunci eksternal

- Kunci kriptografi AES 256-bit (256 bit acak). Kunci eksternal harus`AES_256`. `KeySpec`
- Diaktifkan dan tersedia untuk digunakan. Kunci eksternal harus`ENABLED`. `Status`
- Dikonfigurasi untuk enkripsi dan dekripsi. Kunci eksternal harus mencakup `ENCRYPT` dan`DECRYPT`. `KeyUsage`
- Digunakan hanya dengan kunci KMS ini. Masing-masing `KMS key` di penyimpanan kunci eksternal harus dikaitkan dengan kunci eksternal yang berbeda.

  AWS KMSjuga merekomendasikan agar kunci eksternal digunakan secara eksklusif untuk penyimpanan kunci eksternal. Pembatasan ini memudahkan untuk mengidentifikasi dan menyelesaikan masalah dengan kunci.
- Dapat diakses oleh proxy penyimpanan kunci eksternal untuk penyimpanan kunci eksternal.

  Jika proxy penyimpanan kunci eksternal tidak dapat menemukan kunci menggunakan ID kunci eksternal yang ditentukan, `CreateKey` operasi gagal.
- Dapat menangani lalu lintas yang diantisipasi yang Anda gunakan Layanan AWS menghasilkan. AWS KMSmerekomendasikan agar kunci eksternal disiapkan untuk menangani hingga 1800 permintaan per detik.

Buat kunci KMS di toko kunci eksternal (konsol)

Ada dua cara untuk membuat kunci KMS di toko kunci eksternal.

- Metode 1 (disarankan): Pilih penyimpanan kunci eksternal, lalu buat kunci KMS di toko kunci eksternal itu.
- Metode 2: Buat kunci KMS, lalu tunjukkan bahwa itu ada di toko kunci eksternal.

Jika Anda menggunakan Metode 1, di mana Anda memilih penyimpanan kunci eksternal sebelum Anda membuat kunci Anda, AWS KMS memilih semua properti kunci KMS yang diperlukan untuk Anda dan mengisi ID penyimpanan kunci eksternal Anda. Metode ini menghindari kesalahan yang mungkin Anda buat saat membuat kunci KMS Anda.

> **Note**
>
> Jangan sertakan informasi rahasia atau sensitif dalam alias, deskripsi, atau tag. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Metode 1 (disarankan): Mulai di toko kunci eksternal Anda

Untuk menggunakan metode ini, pilih toko kunci eksternal Anda, lalu buat kunci KMS. AWS KMSKonsol memilih semua properti yang diperlukan untuk Anda dan mengisi ID penyimpanan kunci eksternal Anda. Metode ini menghindari banyak kesalahan yang mungkin Anda buat saat membuat kunci KMS Anda.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Toko kunci khusus, Toko kunci eksternal.
4. Pilih nama toko kunci eksternal Anda.
5. Di pojok kanan atas, pilih Buat kunci KMS di toko kunci ini.

   Jika toko kunci eksternal tidak terhubung, Anda akan diminta untuk menghubungkannya. Jika upaya koneksi gagal, Anda harus menyelesaikan masalah dan menghubungkan penyimpanan kunci eksternal sebelum Anda dapat membuat kunci KMS baru di dalamnya.

   Jika penyimpanan kunci eksternal terhubung, Anda diarahkan ke halaman kunci yang dikelola Pelanggan untuk membuat kunci. Nilai konfigurasi Kunci yang diperlukan sudah dipilih untuk Anda. Juga, ID penyimpanan kunci khusus dari toko kunci eksternal Anda diisi, meskipun Anda dapat mengubahnya.
6. Masukkan ID kunci dari kunci eksternal di pengelola kunci eksternal Anda. Kunci eksternal ini harus memenuhi persyaratan untuk digunakan dengan kunci KMS. Anda tidak dapat mengubah nilai ini setelah kunci dibuat.

Jika kunci eksternal memiliki beberapa ID, masukkan ID kunci yang digunakan proxy penyimpanan kunci eksternal untuk mengidentifikasi kunci eksternal.

7. Konfirmasikan bahwa Anda bermaksud membuat kunci KMS di toko kunci eksternal yang ditentukan.
8. Pilih Berikutnya.

   Sisa dari prosedur ini sama dengan membuat kunci KMS standar.
9. Ketik alias (wajib) dan deskripsi (opsional) untuk kunci KMS.
10. (Opsional). Pada halaman Tambah Tag, tambahkan tag yang mengidentifikasi atau mengkategorikan kunci KMS Anda.

    Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS
11. Pilih Berikutnya.
12. Di bagian Administrator Kunci, pilih pengguna IAM dan peran yang dapat mengelola kunci KMS. Untuk informasi selengkapnya, lihat Mengizinkan administrator kunci mengelola kunci KMS.

    > **Note**
    >
    > Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.
    > Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.
13. (Opsional) Untuk mencegah administrator kunci ini menghapus kunci KMS ini, hapus kotak centang Izinkan administrator kunci untuk menghapus kunci ini.

    Menghapus kunci KMS adalah operasi destruktif dan ireversibel yang dapat membuat ciphertext tidak dapat dipulihkan. Anda tidak dapat membuat ulang kunci KMS simetris di penyimpanan kunci eksternal, bahkan jika Anda memiliki materi kunci eksternal. Namun, menghapus kunci KMS tidak berpengaruh pada kunci eksternal yang terkait. Untuk informasi tentang menghapus

kunci KMS dari penyimpanan kunci eksternal, lihat. Penjadwalan penghapusan kunci KMS dari toko kunci eksternal

14. Pilih Berikutnya.
15. Di bagian Akun ini, pilih pengguna IAM dan peran dalam hal ini Akun AWS yang dapat menggunakan kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Mengizinkan pengguna kunci menggunakan kunci KMS.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.
> Praktik terbaik IAM mencegah penggunaan pengguna IAM dengan kredensil jangka panjang. Bila memungkinkan, gunakan peran IAM, yang menyediakan kredensi sementara. Untuk detailnya, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

16. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, di Akun AWS bagian Lain di bagian bawah halaman, pilih Tambahkan yang lain Akun AWS dan masukkan Akun AWS ID akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Administrator dari pihak lain juga Akun AWS harus mengizinkan akses ke kunci KMS dengan membuat kebijakan IAM untuk pengguna mereka. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

17. Pilih Selanjutnya.
18. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
19. Setelah selesai, pilih Selesai untuk membuat kunci.

Metode 2: Mulai di kunci yang dikelola Pelanggan

Prosedur ini sama dengan prosedur untuk membuat kunci enkripsi simetris dengan bahan AWS KMS kunci. Namun, dalam prosedur ini, Anda menentukan ID penyimpanan kunci khusus dari

penyimpanan kunci eksternal dan ID kunci dari kunci eksternal. Anda juga harus menentukan nilai properti yang diperlukan untuk kunci KMS di penyimpanan kunci eksternal, seperti spesifikasi kunci dan penggunaan kunci.

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih Buat kunci.
5. Pilih Simetris.
6. Dalam Penggunaan kunci, opsi Enkripsi dan dekripsi dipilih untuk Anda. Jangan mengubahnya.
7. Pilih Opsi lanjutan.
8. Untuk asal bahan utama, pilih Toko kunci eksternal.
9. Konfirmasikan bahwa Anda bermaksud membuat kunci KMS di toko kunci eksternal yang ditentukan.
10. Pilih Berikutnya.
11. Pilih baris yang mewakili penyimpanan kunci eksternal untuk kunci KMS baru Anda.

    Anda tidak dapat memilih toko kunci eksternal yang terputus. Untuk menghubungkan toko kunci yang terputus, pilih nama toko kunci, dan kemudian, dari tindakan Key Store, pilih, Connect. Untuk detailnya, lihat Hubungkan toko kunci eksternal (konsol).
12. Masukkan ID kunci dari kunci eksternal di pengelola kunci eksternal Anda. Kunci eksternal ini harus memenuhi persyaratan untuk digunakan dengan kunci KMS. Anda tidak dapat mengubah nilai ini setelah kunci dibuat.

    Jika kunci eksternal memiliki beberapa ID, masukkan ID kunci yang digunakan proxy penyimpanan kunci eksternal untuk mengidentifikasi kunci eksternal.
13. Pilih Berikutnya.

    Sisa dari prosedur ini sama dengan membuat kunci KMS standar.
14. Ketik alias dan deskripsi opsional untuk kunci KMS.
15. (Opsional). Pada halaman Tambah Tag, tambahkan tag yang mengidentifikasi atau mengkategorikan kunci KMS Anda.

Saat Anda menambahkan tanda ke sumber daya AWS Anda, AWS membuat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan oleh tanda. Tag juga dapat digunakan untuk mengontrol akses ke kunci KMS. Untuk informasi tentang menandai kunci KMS, lihat Tombol penandaan dan. ABAC untuk AWS KMS

16. Pilih Berikutnya.
17. Di bagian Administrator Kunci, pilih pengguna IAM dan peran yang dapat mengelola kunci KMS. Untuk informasi selengkapnya, lihat Mengizinkan administrator kunci mengelola kunci KMS.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.

18. (Opsional) Untuk mencegah administrator kunci ini menghapus kunci KMS ini, hapus kotak centang Izinkan administrator kunci untuk menghapus kunci ini.

    Menghapus kunci KMS adalah operasi destruktif dan ireversibel yang dapat membuat ciphertext tidak dapat dipulihkan. Anda tidak dapat membuat ulang kunci KMS simetris di penyimpanan kunci eksternal, bahkan jika Anda memiliki materi kunci eksternal. Namun, menghapus kunci KMS tidak berpengaruh pada kunci eksternal yang terkait. Untuk informasi tentang menghapus kunci KMS dari penyimpanan kunci eksternal, lihat. Penjadwalan penghapusan kunci KMS dari toko kunci eksternal
19. Pilih Berikutnya.
20. Di bagian Akun ini, pilih pengguna IAM dan peran dalam hal ini Akun AWS yang dapat menggunakan kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Mengizinkan pengguna kunci menggunakan kunci KMS.

> **Note**
>
> Kebijakan IAM dapat memberikan izin kepada pengguna dan peran IAM lainnya untuk menggunakan kunci KMS.

21. (Opsional) Anda dapat mengizinkan orang lain Akun AWS untuk menggunakan kunci KMS ini untuk operasi kriptografi. Untuk melakukannya, di Akun AWS bagian Lain di bagian bawah halaman, pilih Tambahkan yang lain Akun AWS dan masukkan Akun AWS ID akun eksternal. Untuk menambahkan beberapa akun eksternal, ulangi langkah ini.

> **Note**
>
> Administrator dari pihak lain juga Akun AWS harus mengizinkan akses ke kunci KMS dengan membuat kebijakan IAM untuk pengguna mereka. Untuk informasi selengkapnya, lihat Memungkinkan pengguna di akun lain untuk menggunakan kunci KMS.

22. Pilih Selanjutnya.
23. Tinjau pengaturan kunci yang Anda pilih. Anda masih bisa kembali dan mengubah semua pengaturan.
24. Setelah selesai, pilih Selesai untuk membuat kunci.

Ketika prosedur berhasil, tampilan menunjukkan kunci KMS baru di toko kunci eksternal yang Anda pilih. Ketika Anda memilih nama atau alias kunci KMS baru, tab konfigurasi Kriptografi pada halaman detailnya menampilkan asal kunci KMS (penyimpanan kunci eksternal), nama, ID, dan jenis penyimpanan kunci kustom, dan ID, penggunaan kunci, dan status kunci eksternal. Jika prosedur gagal, muncul pesan kesalahan yang menjelaskan kegagalan. Untuk, lihatMemecahkan masalah toko kunci eksternal.

> **Tip**
>
> Untuk mempermudah mengidentifikasi kunci KMS di toko kunci kustom, pada halaman Kunci yang dikelola Pelanggan, tambahkan kolom ID penyimpanan kunci Asal dan Kustom ke tampilan. Untuk mengubah bidang tabel, pilih ikon roda gigi di sudut kanan atas halaman. Untuk detailnya, lihat Menyesuaikan tabel kunci KMS Anda.

Buat kunci KMS di penyimpanan kunci eksternal (AWS KMSAPI)

Untuk membuat kunci KMS baru di toko kunci eksternal, gunakan CreateKeyoperasi. Parameter-parameter berikut diperlukan:

- Nilai `Origin` haruslah `EXTERNAL_KEY_STORE`.
- `CustomKeyStoreId`Parameter mengidentifikasi penyimpanan kunci eksternal Anda. Penyimpanan kunci eksternal yang ditentukan harus`CONNECTED`. `ConnectionState`

Untuk menemukan `CustomKeyStoreId` dan`ConnectionState`, gunakan `DescribeCustomKeyStores` operasi.

- `XksKeyId`Parameter mengidentifikasi kunci eksternal. Kunci eksternal ini harus memenuhi persyaratan untuk asosiasi dengan kunci KMS.

Anda juga dapat menggunakan salah satu parameter opsional `CreateKey` operasi, seperti menggunakan parameter `Policy` atau Tag.

> **Note**
>
> Jangan sertakan informasi rahasia atau sensitif di `Tags` bidang `Description` atau bidang. Bidang ini mungkin muncul dalam teks biasa di CloudTrail log dan output lainnya.

Contoh dalam bagian ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung.

Perintah contoh ini menggunakan CreateKeyoperasi untuk membuat kunci KMS di toko kunci eksternal. Respons mencakup properti kunci KMS, ID penyimpanan kunci eksternal, dan ID, penggunaan, dan status kunci eksternal. Untuk informasi rinci tentang bidang-bidang ini, lihatMelihat tombol KMS di toko kunci eksternal.

Sebelum menjalankan perintah ini, ganti contoh ID penyimpanan kunci kustom dengan ID yang valid.

```
$ 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"
    }
  }
}
```

Melihat tombol KMS di toko kunci eksternal

Untuk melihat kunci KMS di toko kunci eksternal, gunakan AWS KMS konsol atau DescribeKeyoperasi. Anda dapat menggunakan teknik yang sama yang akan Anda gunakan untuk melihat kunci yang dikelola AWS KMS pelanggan. Untuk mempelajari dasar-dasarnya, lihat Melihat kunci.

Di AWS KMS konsol, kunci KMS di toko kunci eksternal Anda ditampilkan di halaman kunci yang dikelola Pelanggan, bersama dengan semua kunci yang dikelola pelanggan lainnya di wilayah Anda Akun AWS dan wilayah. Untuk mengidentifikasi kunci KMS di penyimpanan kunci eksternal, filter berdasarkan nilai asal yang khas, penyimpanan kunci eksternal, dan ID penyimpanan kunci kustom.

Lihat informasi selengkapnya di Melihat toko kunci eksternal, Memantau toko kunci eksternal, dan Logging panggilan AWS KMS API dengan AWS CloudTrail.

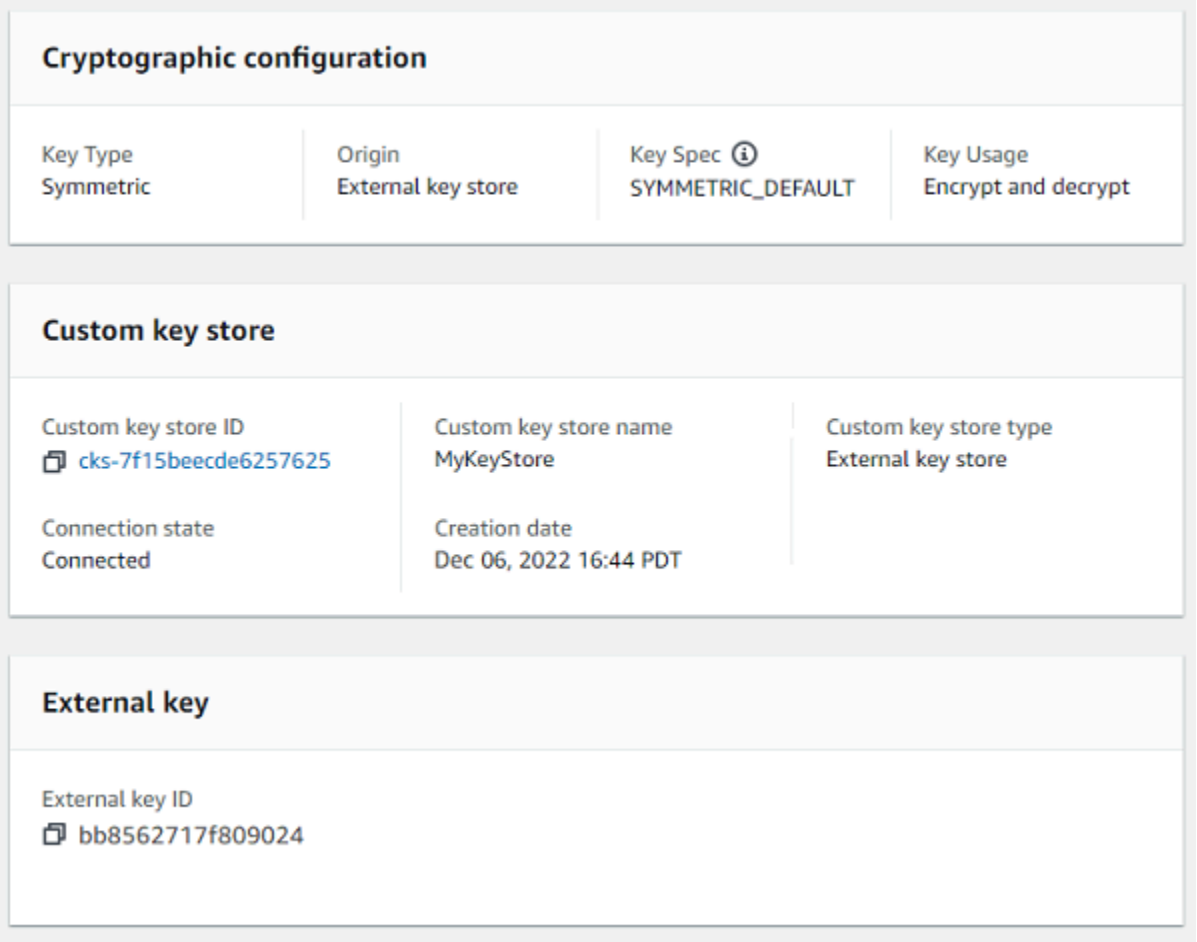
Topik

- Properti kunci KMS di toko kunci eksternal
- Melihat tombol KMS di toko kunci eksternal (konsol)
- Melihat kunci KMS di toko kunci eksternal (AWS KMSAPI)

Properti kunci KMS di toko kunci eksternal

Seperti semua kunci KMS, kunci KMS di penyimpanan kunci eksternal, memiliki ARN kunci, spesifikasi kunci, dan nilai penggunaan kunci, tetapi mereka juga memiliki properti dan nilai properti khusus untuk kunci KMS di toko kunci eksternal. Misalnya, nilai Asal untuk semua kunci KMS di toko kunci eksternal adalah penyimpanan kunci Eksternal.

Untuk kunci KMS di penyimpanan kunci eksternal, tab konfigurasi Kriptografi di AWS KMS konsol mencakup dua bagian tambahan, penyimpanan kunci khusus dan kunci Eksternal.



Properti toko kunci kustom

Nilai-nilai berikut muncul di bagian penyimpanan kunci kustom pada tab konfigurasi Kriptografi dan dalam DescribeKeyrespons. Properti ini berlaku untuk semua toko kunci khusus, termasuk toko AWS CloudHSM kunci dan toko kunci eksternal.

ID penyimpanan kunci kustom

ID unik yang ditetapkan AWS KMS ke toko kunci kustom.

Nama penyimpanan kunci kustom

Nama ramah yang Anda tetapkan ke toko kunci kustom saat Anda membuatnya. Anda dapat mengubah nilai ini kapan saja.

Jenis toko kunci khusus

Jenis toko kunci khusus. Nilai yang valid adalah AWS CloudHSM (`AWS_CLOUDHSM`) atau External key store (`EXTERNAL_KEY_STORE`). Anda tidak dapat mengubah jenis setelah Anda membuat toko kunci kustom.

Tanggal pembuatan

Tanggal penyimpanan kunci khusus dibuat. Tanggal ini ditampilkan dalam waktu setempat untukWilayah AWS.

Status koneksi

Menunjukkan apakah toko kunci khusus terhubung ke toko kunci pendukungnya. Status koneksi `DISCONNECTED` hanya jika toko kunci khusus tidak pernah terhubung ke toko kunci pendukungnya, atau sengaja terputus. Untuk detailnya, lihat the section called “Status koneksi”.

Properti kunci eksternal

Properti kunci eksternal muncul di bagian kunci eksternal dari tab konfigurasi Kriptografi dan dalam `XksKeyConfiguration` elemen DescribeKeyrespons.

Bagian kunci Eksternal muncul di AWS KMS konsol hanya untuk kunci KMS di toko kunci eksternal. Ini memberikan informasi tentang kunci eksternal yang terkait dengan kunci KMS. Kunci eksternal adalah kunci kriptografi di luar AWS yang berfungsi sebagai bahan kunci untuk kunci KMS di toko kunci eksternal. Ketika Anda mengenkripsi atau mendekripsi dengan kunci KMS, operasi dilakukan oleh manajer kunci eksternal Anda menggunakan kunci eksternal yang ditentukan.

Nilai-nilai berikut muncul di bagian kunci Eksternal.

ID kunci eksternal

Pengidentifikasi untuk kunci eksternal di manajer kunci eksternalnya. Ini adalah nilai yang digunakan proxy penyimpanan kunci eksternal untuk mengidentifikasi kunci eksternal. Anda menentukan ID kunci eksternal saat Anda membuat kunci KMS dan Anda tidak dapat mengubahnya. Jika nilai ID kunci eksternal yang Anda gunakan untuk membuat kunci KMS berubah atau menjadi tidak valid, Anda harus menjadwalkan kunci KMS untuk dihapus dan membuat kunci KMS baru dengan nilai ID kunci eksternal yang benar.

Melihat tombol KMS di toko kunci eksternal (konsol)

Untuk melihat kunci KMS di toko kunci eksternal (Konsol)

1. Buka konsol AWS KMS di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilihan Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Untuk mengidentifikasi kunci KMS di toko kunci eksternal Anda, tambahkan kolom ID penyimpanan kunci Asal dan Kustom ke tabel kunci Anda. Kunci KMS di setiap toko kunci eksternal memiliki nilai Asal penyimpanan kunci Eksternal.

   Di sudut kanan atas, pilih ikon roda gigi, pilih ID penyimpanan kunci Origin dan Kustom, lalu pilih Konfirmasi.
5. Pilih alias atau ID kunci kunci KMS di penyimpanan kunci eksternal.
6. Untuk melihat properti khusus untuk kunci KMS di penyimpanan kunci eksternal, pilih tab konfigurasi Kriptografi. Nilai khusus untuk kunci KMS di toko kunci eksternal muncul di toko kunci kustom dan bagian kunci Eksternal.

Melihat kunci KMS di toko kunci eksternal (AWS KMSAPI)

Untuk melihat kunci KMS di penyimpanan kunci eksternal (API)

Anda menggunakan operasi AWS KMS API yang sama untuk melihat kunci KMS di penyimpanan kunci eksternal yang akan Anda gunakan untuk kunci KMS apa pun, termasuk, ListKeysDescribeKey, dan. GetKeyPolicy Misalnya, `describe-key` operasi berikut di AWS CLI menunjukkan bidang khusus untuk kunci KMS di toko kunci eksternal. Sebelum menjalankan perintah seperti ini, ganti contoh ID kunci KMS dengan nilai yang valid.

```
$ 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"
    }
  }
}
```

Menggunakan kunci KMS di toko kunci eksternal

Setelah Anda membuat kunci KMS enkripsi simetris di penyimpanan kunci eksternal, Anda dapat menggunakannya untuk operasi kriptografi berikut:

- Enkripsi
- Dekripsi
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

Operasi enkripsi simetris yang menghasilkan pasangan kunci data asimetris, GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintext, tidak didukung di penyimpanan kunci khusus.

Konteks enkripsi didukung untuk semua operasi kriptografi dengan kunci KMS di penyimpanan kunci eksternal. Seperti biasa, menggunakan konteks enkripsi adalah praktik terbaik keamanan yang AWS KMS merekomendasikan.

Ketika Anda menggunakan kunci KMS Anda dalam permintaan, identifikasi kunci KMS dengan ID kunci, kunci ARN, alias, atau alias ARN. Anda tidak perlu menentukan toko kunci eksternal. Respons mencakup bidang yang sama yang dikembalikan untuk kunci KMS enkripsi simetris apa pun. Namun,

ketika Anda menggunakan kunci KMS di penyimpanan kunci eksternal, operasi enkripsi dan dekripsi dilakukan oleh manajer kunci eksternal Anda menggunakan kunci eksternal yang terkait dengan kunci KMS.

Untuk memastikan bahwa ciphertext yang dienkripsi oleh kunci KMS di penyimpanan kunci eksternal setidaknya seaman ciphertext apa pun yang dienkripsi oleh kunci KMS standar, gunakan enkripsi ganda. AWS KMS Data pertama kali dienkripsi dalam AWS KMS menggunakan materi AWS KMS kunci. Kemudian dienkripsi oleh manajer kunci eksternal Anda menggunakan kunci eksternal untuk kunci KMS. Untuk mendekripsi ciphertext terenkripsi ganda, ciphertext pertama kali didekripsi oleh pengelola kunci eksternal Anda menggunakan kunci eksternal untuk kunci KMS. Kemudian didekripsi dalam AWS KMS menggunakan bahan AWS KMS kunci untuk kunci KMS.

Untuk memungkinkan ini terjadi, syarat-syarat berikut diperlukan.

- Status kunci dari kunci KMS harus`Enabled`. Untuk menemukan status kunci, lihat bidang Status untuk kunci terkelola pelanggan pada AWS KMSkonsol atau `KeyState` bidang dalam DescribeKeyrespons.
- Penyimpanan kunci eksternal yang menampung kunci KMS harus terhubung ke proxy penyimpanan kunci eksternalnya, yaitu, status koneksi penyimpanan kunci eksternal harus`CONNECTED`.

  Anda dapat melihat status koneksi pada halaman penyimpanan kunci eksternal di AWS KMS konsol atau dalam DescribeCustomKeyStoresrespons. Status koneksi penyimpanan kunci eksternal juga ditampilkan pada halaman detail untuk kunci KMS di AWS KMS konsol. Pada halaman detail, pilih tab Konfigurasi kriptografi dan lihat bidang Status koneksi di bagian Penyimpanan kunci khusus.

  Jika status koneksi`DISCONNECTED`, Anda harus menghubungkannya terlebih dahulu. Jika status koneksi`FAILED`, Anda harus menyelesaikan masalah, lepaskan penyimpanan kunci eksternal, dan kemudian hubungkan. Untuk instruksi, lihatMenghubungkan dan memutuskan penyimpanan kunci eksternal..
- Proxy penyimpanan kunci eksternal harus dapat menemukan kunci eksternal.
- Kunci eksternal harus diaktifkan dan harus melakukan enkripsi dan dekripsi.

  Status kunci eksternal tidak tergantung dan tidak terpengaruh oleh perubahan status kunci kunci KMS, termasuk mengaktifkan dan menonaktifkan kunci KMS. Demikian pula, menonaktifkan atau menghapus kunci eksternal tidak mengubah status kunci dari kunci KMS, tetapi operasi kriptografi menggunakan kunci KMS terkait akan gagal.

Jika kondisi ini tidak terpenuhi, operasi kriptografi gagal, dan AWS KMS mengembalikan pengecualian `KMSInvalidStateException`. Anda mungkin perlu menghubungkan kembali penyimpanan kunci eksternal atau menggunakan alat pengelola kunci eksternal Anda untuk mengkonfigurasi ulang atau memperbaiki kunci eksternal Anda. Untuk bantuan tambahan, lihat the section called “Memecahkan masalah toko kunci eksternal”.

Saat menggunakan kunci KMS di penyimpanan kunci eksternal, ketahuilah bahwa kunci KMS di setiap toko kunci eksternal berbagi kuota permintaan toko kunci kustom untuk operasi kriptografi. Jika Anda melebihi kuota, AWS KMS mengembalikan `ThrottlingException`. Untuk detail tentang kuota permintaan toko kunci kustom, lihatKuota permintaan toko kunci kustom.

Penjadwalan penghapusan kunci KMS dari toko kunci eksternal

Ketika Anda yakin bahwa Anda tidak perlu menggunakan untuk operasi kriptografi apa pun, Anda dapat menjadwalkan penghapusan kunci KMS. AWS KMS key Gunakan prosedur yang sama yang akan Anda gunakan untuk menjadwalkan penghapusan kunci KMS apa pun. AWS KMS Menghapus kunci KMS dari penyimpanan kunci eksternal tidak berpengaruh pada kunci eksternal yang berfungsi sebagai bahan utamanya.

Anda dapat membatalkan penghapusan kunci KMS yang dijadwalkan selama masa tunggu wajibnya. Namun, kunci KMS yang dihapus tidak dapat dipulihkan. Anda tidak dapat membuat ulang kunci KMS enkripsi simetris di penyimpanan kunci eksternal, bahkan Anda menggunakan kunci eksternal yang sama. Karena setiap kunci KMS simetris di penyimpanan kunci eksternal memiliki bahan kunci dan metadata yang unik, hanya AWS KMS kunci yang mengenkripsi ciphertext simetris yang dapat mendekripsi itu. AWS KMS

> **Warning**
>
> Menghapus kunci KMS adalah operasi yang merusak dan berpotensi berbahaya yang mencegah Anda memulihkan semua data yang dienkripsi di bawah kunci KMS. Sebelum menjadwalkan penghapusan kunci KMS, periksa penggunaan sebelumnya dari kunci KMS dan buat CloudWatch alarm Amazon yang memberi tahu Anda ketika seseorang mencoba menggunakan kunci KMS saat sedang menunggu penghapusan. Jika memungkinkan, nonaktifkan kunci KMS, alih-alih menghapusnya.

Saat Anda menjadwalkan penghapusan kunci KMS dari penyimpanan kunci eksternal, status kuncinya berubah menjadi penghapusan Tertunda. Kunci KMS tetap dalam status penghapusan Tertunda selama periode tunggu, bahkan jika kunci KMS menjadi tidak tersedia karena Anda

telah memutuskan penyimpanan kunci eksternal. Ini memungkinkan Anda untuk membatalkan penghapusan kunci KMS kapan saja selama masa tunggu. Ketika masa tunggu berakhir, AWS KMS hapus kunci KMS dari. AWS KMS

Ketika Anda menjadwalkan penghapusan kunci KMS dari penyimpanan kunci eksternal, kunci KMS menjadi tidak dapat digunakan segera (tergantung pada konsistensi akhirnya). Namun, sumber daya yang dienkripsi dengan kunci data yang dilindungi oleh kunci KMS tidak terpengaruh sampai kunci KMS digunakan lagi, seperti untuk mendekripsi kunci data. Masalah ini memengaruhiLayanan AWS, banyak di antaranya menggunakan kunci data untuk melindungi sumber daya Anda. Untuk detailnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data.

Anda dapat memantau penjadwalan, pembatalan, dan penghapusan kunci KMS di log Anda. AWS CloudTrail

Memecahkan masalah toko kunci eksternal

Resolusi untuk sebagian besar masalah dengan penyimpanan kunci eksternal ditunjukkan oleh pesan kesalahan yang AWS KMS ditampilkan dengan setiap pengecualian, atau oleh kode kesalahan koneksi yang AWS KMS kembali ketika upaya untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal gagal. Namun, beberapa masalah sedikit lebih kompleks.

Saat mendiagnosis masalah dengan penyimpanan kunci eksternal, pertama-tama temukan penyebabnya. Ini akan mempersempit berbagai solusi dan membuat pemecahan masalah Anda lebih efisien.

- AWS KMS — Masalahnya mungkin ada di dalam AWS KMS, seperti nilai yang salah dalam konfigurasi penyimpanan kunci eksternal Anda.
- Eksternal — Masalah mungkin berasal dari luar AWS KMS, termasuk masalah dengan konfigurasi atau pengoperasian proxy penyimpanan kunci eksternal, manajer kunci eksternal, kunci eksternal, atau layanan titik akhir VPC.
- Jaringan — Ini mungkin masalah dengan konektivitas atau jaringan, seperti masalah dengan titik akhir proxy, port, atau nama atau domain DNS pribadi Anda.

> **Note**
>
> Ketika operasi manajemen pada penyimpanan kunci eksternal gagal, mereka menghasilkan beberapa pengecualian yang berbeda. Tetapi operasi AWS KMS kriptografi kembali

`KMSInvalidStateException` untuk semua kegagalan yang terkait dengan konfigurasi eksternal atau status koneksi dari penyimpanan kunci eksternal. Untuk mengidentifikasi masalah, gunakan teks pesan kesalahan yang menyertainya. ConnectCustomKeyStoreOperasi berhasil dengan cepat sebelum proses koneksi selesai. Untuk menentukan apakah proses koneksi berhasil, lihat status koneksi penyimpanan kunci eksternal. Jika proses koneksi gagal, AWS KMS mengembalikan kode kesalahan koneksi yang menjelaskan penyebabnya dan menyarankan solusi.

Topik

- Alat pemecahan masalah untuk penyimpanan kunci eksternal
- Kesalahan konfigurasi
- Kesalahan koneksi penyimpanan kunci eksternal
- Kesalahan latensi dan batas waktu
- Kesalahan kredensi otentikasi
- Kesalahan status kunci
- Kesalahan dekripsi
- Kesalahan kunci eksternal
- Masalah proxy
- Masalah otorisasi proxy

Alat pemecahan masalah untuk penyimpanan kunci eksternal

AWS KMS menyediakan beberapa alat untuk membantu Anda mengidentifikasi dan menyelesaikan masalah dengan toko kunci eksternal Anda dan kuncinya. Gunakan alat ini bersama dengan alat yang disediakan dengan proxy penyimpanan kunci eksternal dan manajer kunci eksternal Anda.

Note

Proxy penyimpanan kunci eksternal dan pengelola kunci eksternal Anda mungkin menyediakan metode yang lebih mudah untuk membuat dan memelihara penyimpanan kunci eksternal Anda dan kunci KMS-nya. Untuk detailnya, lihat dokumentasi untuk alat eksternal Anda.

AWS KMS pengecualian dan pesan kesalahan

AWS KMS memberikan pesan kesalahan terperinci tentang masalah apa pun yang dihadapinya. Anda dapat menemukan informasi tambahan tentang AWS KMS pengecualian di Referensi AWS Key Management Service API dan AWS SDK. Bahkan jika Anda menggunakan AWS KMS konsol, Anda mungkin menemukan referensi ini berguna. Misalnya, lihat daftar Kesalahan untuk `CreateCustomKeyStores` operasi.

Jika masalah muncul di AWS layanan yang berbeda, seperti ketika Anda menggunakan kunci KMS di penyimpanan kunci eksternal Anda untuk melindungi sumber daya di AWS layanan lain, AWS layanan mungkin memberikan informasi tambahan untuk membantu Anda mengidentifikasi masalah. Jika AWS layanan tidak menyediakan pesan, Anda dapat melihat pesan kesalahan di CloudTrail log yang merekam penggunaan kunci KMS Anda.

CloudTrail log

Setiap operasi AWS KMS API, termasuk tindakan di AWS KMS konsol, dicatat dalam AWS CloudTrail log. AWS KMS mencatat entri log untuk operasi yang berhasil dan gagal. Untuk operasi yang gagal, entri log menyertakan nama AWS KMS pengecualian (`errorCode`) dan pesan kesalahan (`errorMessage`). Anda dapat menggunakan informasi ini untuk membantu Anda mengidentifikasi dan mengatasi kesalahan. Sebagai contoh, lihat Dekripsi kegagalan dengan kunci KMS di toko kunci eksternal.

Entri log juga menyertakan ID permintaan. Jika permintaan mencapai proxy penyimpanan kunci eksternal Anda, Anda dapat menggunakan ID permintaan di entri log untuk menemukan permintaan yang sesuai di log proxy Anda, jika proxy Anda menyediakannya.

CloudWatch metrik

AWS KMS mencatat CloudWatch metrik Amazon terperinci tentang pengoperasian dan kinerja penyimpanan kunci eksternal Anda, termasuk latensi, pembatasan, kesalahan proxy, status pengelola kunci eksternal, jumlah hari hingga sertifikat TLS Anda kedaluwarsa, dan usia kredensyal autentikasi proxy yang dilaporkan. Anda dapat menggunakan metrik ini untuk mengembangkan model data untuk pengoperasian penyimpanan kunci eksternal dan CloudWatch alarm yang mengingatkan Anda tentang masalah yang akan datang sebelum terjadi.

> **Important**
>
> AWS KMS merekomendasikan agar Anda membuat CloudWatch alarm untuk memantau metrik penyimpanan kunci eksternal. Alarm ini akan mengingatkan Anda tentang tanda-tanda awal masalah sebelum berkembang.

Grafik pemantauan

AWS KMS menampilkan grafik CloudWatch metrik penyimpanan kunci eksternal pada halaman detail untuk setiap penyimpanan kunci eksternal di AWS KMS konsol. Anda dapat menggunakan data dalam grafik untuk membantu menemukan sumber kesalahan, mendeteksi masalah yang akan datang, menetapkan garis dasar, dan memperbaiki ambang alarm Anda. CloudWatch Untuk detail tentang menafsirkan grafik pemantauan dan menggunakan datanya, lihat. Memantau toko kunci eksternal

Menampilkan toko kunci eksternal dan kunci KMS

AWS KMS menampilkan informasi terperinci tentang penyimpanan kunci eksternal Anda dan kunci KMS di toko kunci eksternal di AWS KMS konsol, dan dalam respons terhadap DescribeCustomKeyStoresdan DescribeKeyoperasi. Tampilan ini mencakup bidang khusus untuk penyimpanan kunci eksternal dan kunci KMS dengan informasi yang dapat Anda gunakan untuk pemecahan masalah, seperti status koneksi penyimpanan kunci eksternal dan ID kunci eksternal yang terkait dengan kunci KMS. Untuk detailnya, lihat Melihat toko kunci eksternal dan Melihat tombol KMS di toko kunci eksternal.

Klien Uji Proxy XKS

AWS KMS menyediakan klien pengujian open source yang memverifikasi bahwa proxy penyimpanan kunci eksternal Anda sesuai dengan Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal. Anda dapat menggunakan klien pengujian ini untuk mengidentifikasi dan menyelesaikan masalah dengan proxy penyimpanan kunci eksternal Anda.

Kesalahan konfigurasi

Saat membuat penyimpanan kunci eksternal, Anda menentukan nilai properti yang terdiri dari konfigurasi penyimpanan kunci eksternal Anda, seperti kredensi otentikasi proxy, titik akhir URI proxy, jalur URI proxy, dan nama layanan titik akhir VPC. Ketika AWS KMS mendeteksi kesalahan dalam nilai properti, operasi gagal dan mengembalikan kesalahan yang menunjukkan nilai yang salah.

Banyak masalah konfigurasi dapat diselesaikan dengan memperbaiki nilai yang salah. Anda dapat memperbaiki jalur URI proxy yang tidak valid atau kredensi otentikasi proxy tanpa memutuskan penyimpanan kunci eksternal. Untuk definisi nilai-nilai ini, termasuk persyaratan keunikan, lihat. Memasang prasyarat Untuk petunjuk tentang memperbarui nilai-nilai ini, lihatMengedit properti penyimpanan kunci eksternal.

Untuk menghindari kesalahan pada jalur URI proxy dan nilai kredensi autentikasi proxy, saat membuat atau memperbarui penyimpanan kunci eksternal, unggah file konfigurasi proxy ke konsol. AWS KMS Ini adalah file berbasis JSON dengan jalur URI proxy dan nilai kredensi otentikasi proxy yang disediakan oleh proxy penyimpanan kunci eksternal atau pengelola kunci eksternal Anda. Anda tidak dapat menggunakan file konfigurasi proxy dengan operasi AWS KMS API, tetapi Anda dapat menggunakan nilai dalam file untuk membantu Anda memberikan nilai parameter untuk permintaan API yang cocok dengan nilai dalam proxy Anda.

Kesalahan konfigurasi umum

Pengecualian: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (operasi kriptografi), `XksProxyInvalidConfigurationException` (operasi manajemen, kecuali untuk) `CreateKey`

Kode kesalahan koneksi:`XKS_PROXY_INVALID_CONFIGURATION`, `XKS_PROXY_INVALID_TLS_CONFIGURATION`

Untuk penyimpanan kunci eksternal dengan konektivitas titik akhir publik, AWS KMS uji nilai properti saat Anda membuat dan memperbarui penyimpanan kunci eksternal. Untuk penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC, AWS KMS uji nilai properti saat Anda menghubungkan dan memperbarui penyimpanan kunci eksternal.

> **Note**
>
> `ConnectCustomKeyStore`Operasi, yang asinkron, mungkin berhasil meskipun upaya untuk menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternalnya gagal. Dalam hal ini, tidak ada pengecualian, tetapi status koneksi penyimpanan kunci eksternal Gagal, dan kode kesalahan koneksi menjelaskan pesan kesalahan. Untuk informasi selengkapnya, lihat Kesalahan koneksi penyimpanan kunci eksternal.

Jika AWS KMS mendeteksi kesalahan dalam nilai properti, operasi gagal dan kembali `XksProxyInvalidConfigurationException` dengan salah satu pesan kesalahan berikut.

Proxy penyimpanan kunci eksternal menolak permintaan karena jalur URI tidak valid. Verifikasi jalur URI untuk penyimpanan kunci eksternal Anda dan perbarui jika perlu.

- Jalur URI proxy adalah jalur dasar untuk AWS KMS permintaan ke API proxy. Jika jalur ini salah, semua permintaan ke proxy gagal. Untuk melihat jalur URI proxy saat ini untuk penyimpanan kunci eksternal Anda, gunakan AWS KMS konsol atau `DescribeCustomKeyStores` operasi. Untuk menemukan jalur URI proxy yang benar, lihat dokumentasi proxy penyimpanan kunci eksternal Anda. Untuk bantuan mengoreksi nilai jalur URI proxy Anda, lihatMengedit properti penyimpanan kunci eksternal.
- Jalur URI proxy untuk proxy penyimpanan kunci eksternal Anda dapat berubah dengan pembaruan ke proxy penyimpanan kunci eksternal atau pengelola kunci eksternal Anda. Untuk informasi tentang perubahan ini, lihat dokumentasi untuk proxy penyimpanan kunci eksternal atau pengelola kunci eksternal.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`
AWS KMS tidak dapat membuat koneksi TLS ke proxy penyimpanan kunci eksternal. Verifikasi konfigurasi TLS, termasuk sertifikatnya.

- Semua proxy penyimpanan kunci eksternal memerlukan sertifikat TLS. Sertifikat TLS harus dikeluarkan oleh otoritas sertifikat publik (CA) yang didukung untuk toko kunci eksternal. Untuk daftar CA yang didukung, lihat Otoritas Sertifikat Tepercaya di Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal.
- Untuk konektivitas titik akhir publik, nama umum subjek (CN) pada sertifikat TLS harus cocok dengan nama domain di titik akhir URI proxy untuk proxy penyimpanan kunci eksternal. Misalnya, jika titik akhir publik adalah https://myproxy.xks.example.com, TLS, CN pada sertifikat TLS harus atau. `myproxy.xks.example.com` `*.xks.example.com`
- Untuk konektivitas layanan titik akhir VPC, nama umum subjek (CN) pada sertifikat TLS harus cocok dengan nama DNS pribadi untuk layanan titik akhir VPC Anda. Misalnya, jika nama DNS pribadi adalah myproxy-private.xks.example.com, CN pada sertifikat TLS harus atau. `myproxy-private.xks.example.com` `*.xks.example.com`
- Sertifikat TLS tidak dapat kedaluwarsa. Untuk mendapatkan tanggal kedaluwarsa sertifikat TLS, gunakan alat SSL, seperti OpenSSL. Untuk memantau tanggal kedaluwarsa sertifikat TLS yang terkait dengan penyimpanan kunci eksternal, gunakan metrik. XksProxyCertificateDaysToExpire

CloudWatch Jumlah hari untuk tanggal kedaluwarsa sertifikasi TLS Anda juga muncul di bagian Pemantauan konsol. AWS KMS

- Jika Anda menggunakan konektivitas titik akhir publik, gunakan alat uji SSL untuk menguji konfigurasi SSL Anda. Kesalahan koneksi TLS dapat terjadi akibat rantai sertifikat yang salah.

Kesalahan konfigurasi konektivitas layanan titik akhir VPC

Pengecualian:`XksProxyVpcEndpointServiceNotFoundException`, `XksProxyVpcEndpointServiceInvalidConfigurationException`

Selain masalah konektivitas umum, Anda mungkin mengalami masalah berikut saat membuat, menghubungkan, atau memperbarui penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC. AWS KMS menguji nilai properti penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC saat membuat, menghubungkan, dan memperbarui penyimpanan kunci eksternal. Ketika operasi manajemen gagal karena kesalahan konfigurasi, mereka menghasilkan pengecualian berikut:

```
XksProxyVpcEndpointServiceNotFoundException
```

Penyebabnya mungkin salah satu dari berikut ini:

- Nama layanan titik akhir VPC yang salah. Verifikasi bahwa nama layanan titik akhir VPC untuk penyimpanan kunci eksternal sudah benar dan cocok dengan nilai titik akhir URI proxy untuk penyimpanan kunci eksternal. Untuk menemukan nama layanan titik akhir VPC, gunakan konsol VPC Amazon atau operasinya. DescribeVpcEndpointServices Untuk menemukan nama layanan titik akhir VPC dan titik akhir URI proxy dari penyimpanan kunci eksternal yang ada, gunakan AWS KMS konsol atau operasi. DescribeCustomKeyStores Lihat perinciannya di Melihat toko kunci eksternal.
- Layanan titik akhir VPC mungkin berbeda Wilayah AWS dari penyimpanan kunci eksternal. Verifikasi bahwa layanan titik akhir VPC dan penyimpanan kunci eksternal berada di Wilayah yang sama. (Nama eksternal nama Wilayah, seperti, adalah bagian dari nama layanan titik akhir VPC`us-east-1`, seperti com.amazonaws.vpce.us-east-1. vpce-svc-example.) Untuk daftar persyaratan layanan titik akhir VPC untuk penyimpanan kunci eksternal, lihat. Layanan titik akhir VPC Anda tidak dapat memindahkan layanan titik akhir VPC atau penyimpanan kunci eksternal ke Wilayah lain. Namun, Anda dapat membuat penyimpanan kunci eksternal baru di Wilayah yang

sama dengan layanan titik akhir VPC. Untuk detailnya, lihat Mengkonfigurasi konektivitas layanan titik akhir VPC dan Membuat toko kunci eksternal.

- AWS KMS bukan prinsipal yang diizinkan untuk layanan titik akhir VPC. Daftar Allow principals untuk layanan endpoint VPC harus menyertakan nilai, seperti. `cks.kms.<region>.amazonaws.com` `cks.kms.eu-west-3.amazonaws.com` Untuk petunjuk tentang menambahkan nilai ini, lihat Mengelola izin di AWS PrivateLink Panduan.

XksProxyVpcEndpointServiceInvalidConfigurationException

Kesalahan ini terjadi ketika layanan titik akhir VPC gagal memenuhi salah satu persyaratan berikut:

- VPC membutuhkan setidaknya dua subnet pribadi, masing-masing di Availability Zone yang berbeda. Untuk bantuan menambahkan subnet ke VPC Anda, lihat Membuat subnet di VPC Anda di Panduan Pengguna Amazon VPC.
- Jenis layanan titik akhir VPC Anda harus menggunakan penyeimbang beban jaringan, bukan penyeimbang beban gateway.
- Penerimaan tidak harus diperlukan untuk layanan titik akhir VPC (Penerimaan yang diperlukan harus salah.). Jika penerimaan manual dari setiap permintaan koneksi diperlukan, AWS KMS tidak dapat menggunakan layanan titik akhir VPC untuk terhubung ke proxy penyimpanan kunci eksternal. Untuk detailnya, lihat Menerima atau menolak permintaan koneksi di AWS PrivateLink Panduan.
- Layanan titik akhir VPC harus memiliki nama DNS pribadi yang merupakan subdomain dari domain publik. Misalnya, jika nama DNS pribadi adalah`https://myproxy-private.xks.example.com`, `example.com` domain `xks.example.com` atau harus memiliki server DNS publik. Untuk melihat atau mengubah nama DNS pribadi untuk layanan titik akhir VPC Anda, lihat Mengelola nama DNS untuk layanan titik akhir VPC di Panduan.AWS PrivateLink
- Status verifikasi Domain domain untuk nama DNS pribadi Anda harus`verified`. Untuk melihat dan memperbarui status verifikasi domain nama DNS pribadi, lihatMemverifikasi domain nama DNS pribadi Anda. Mungkin perlu beberapa menit agar status verifikasi yang diperbarui muncul setelah Anda menambahkan catatan teks yang diperlukan.

> **Note**
>
> Domain DNS pribadi dapat diverifikasi hanya jika itu adalah subdomain dari domain publik. Jika tidak, status verifikasi domain DNS pribadi tidak berubah, bahkan setelah Anda menambahkan catatan TXT yang diperlukan.

- Nama DNS pribadi dari layanan titik akhir VPC harus cocok dengan nilai titik akhir URI proxy untuk penyimpanan kunci eksternal. Untuk penyimpanan kunci eksternal dengan konektivitas layanan titik akhir VPC, titik akhir URI proxy harus `https://` diikuti dengan nama DNS pribadi dari layanan titik akhir VPC. Untuk melihat nilai titik akhir URI proxy, lihatMelihat toko kunci eksternal. Untuk mengubah nilai titik akhir URI proxy, lihatMengedit properti penyimpanan kunci eksternal.

Kesalahan koneksi penyimpanan kunci eksternal

Proses menghubungkan penyimpanan kunci eksternal ke proxy penyimpanan kunci eksternal membutuhkan waktu sekitar lima menit untuk menyelesaikannya. Kecuali gagal dengan cepat, `ConnectCustomKeyStore` operasi mengembalikan respons HTTP 200 dan objek JSON tanpa properti. Namun, respons awal ini tidak menunjukkan bahwa koneksi berhasil. Untuk menentukan apakah penyimpanan kunci eksternal terhubung, lihat status koneksinya. Jika koneksi gagal, status koneksi penyimpanan kunci eksternal berubah `FAILED` dan AWS KMS mengembalikan kode kesalahan koneksi yang menjelaskan penyebab kegagalan.

> **Note**
>
> Saat status koneksi penyimpanan kunci khususFAILED, Anda harus memutuskan penyimpanan kunci khusus sebelum mencoba menghubungkannya kembali. Anda tidak dapat menghubungkan penyimpanan kunci kustom dengan status koneksi `FAILED`.

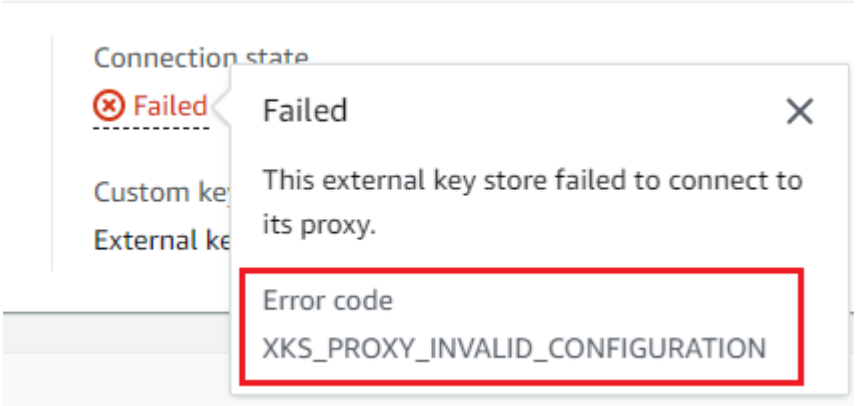
Untuk melihat status koneksi penyimpanan kunci eksternal:

- Dalam DescribeCustomKeyStorestanggapannya, lihat nilai `ConnectionState` elemen.
- Di AWS KMS konsol, status koneksi muncul di tabel penyimpanan kunci eksternal. Juga, pada halaman detail untuk setiap penyimpanan kunci eksternal, status Koneksi muncul di bagian Konfigurasi umum.

Ketika status koneksi`FAILED`, kode kesalahan koneksi membantu menjelaskan kesalahan.

Untuk melihat kode kesalahan koneksi:

- Dalam DescribeCustomKeyStorestanggapannya, lihat nilai `ConnectionErrorCode` elemen. Elemen ini muncul dalam `DescribeCustomKeyStores` respons hanya ketika `ConnectionState` ada`FAILED`.
- Untuk melihat kode kesalahan koneksi di AWS KMS konsol, pada halaman detail untuk penyimpanan kunci eksternal dan arahkan kursor ke nilai Gagal.



Kode kesalahan koneksi untuk penyimpanan kunci eksternal

Kode kesalahan koneksi berikut berlaku untuk toko kunci eksternal

`INTERNAL_ERROR`

AWS KMS tidak dapat menyelesaikan permintaan karena kesalahan internal. Coba lagi permintaannya. Untuk permintaan `ConnectCustomKeyStore`, putuskan koneksi penyimpanan kunci kustom sebelum mencoba menyambungkan lagi.

`INVALID_CREDENTIALS`

Salah satu atau kedua `XksProxyAuthenticationCredential` nilai tidak valid pada proxy penyimpanan kunci eksternal yang ditentukan.

`NETWORK_ERRORS`

Kesalahan jaringan AWS KMS mencegah menghubungkan toko kunci khusus ke toko kunci pendukungnya.

`XKS_PROXY_ACCESS_DENIED`

AWS KMS permintaan ditolak akses ke proxy penyimpanan kunci eksternal. Jika proxy penyimpanan kunci eksternal memiliki aturan otorisasi, verifikasi bahwa mereka mengizinkan AWS KMS untuk berkomunikasi dengan proxy atas nama Anda.

`XKS_PROXY_INVALID_CONFIGURATION`

Kesalahan konfigurasi mencegah penyimpanan kunci eksternal terhubung ke proxy-nya. Verifikasi nilai`XksProxyUriPath`.

`XKS_PROXY_INVALID_RESPONSE`

AWS KMS tidak dapat menafsirkan respons dari proxy penyimpanan kunci eksternal. Jika Anda melihat kode kesalahan koneksi ini berulang kali, beri tahu vendor proxy penyimpanan kunci eksternal Anda.

`XKS_PROXY_INVALID_TLS_CONFIGURATION`

AWS KMS tidak dapat terhubung ke proxy penyimpanan kunci eksternal karena konfigurasi TLS tidak valid. Verifikasi bahwa proxy penyimpanan kunci eksternal mendukung TLS 1.2 atau 1.3. Juga, verifikasi bahwa sertifikat TLS tidak kedaluwarsa, bahwa itu cocok dengan nama host dalam `XksProxyUriEndpoint` nilai, dan bahwa itu ditandatangani oleh otoritas sertifikat tepercaya yang termasuk dalam daftar Otoritas Sertifikat Tepercaya.

`XKS_PROXY_NOT_REACHABLE`

AWS KMS tidak dapat berkomunikasi dengan proxy penyimpanan kunci eksternal Anda. Verifikasi bahwa `XksProxyUriEndpoint` `XksProxyUriPath` dan benar. Gunakan alat untuk proxy penyimpanan kunci eksternal Anda untuk memverifikasi bahwa proxy aktif dan tersedia di jaringannya. Juga, verifikasi bahwa instans pengelola kunci eksternal Anda beroperasi dengan benar. Upaya koneksi gagal dengan kode kesalahan koneksi ini jika proxy melaporkan bahwa semua instance pengelola kunci eksternal tidak tersedia.

`XKS_PROXY_TIMED_OUT`

AWS KMS dapat terhubung ke proxy penyimpanan kunci eksternal, tetapi proxy tidak merespons AWS KMS dalam waktu yang ditentukan. Jika Anda melihat kode kesalahan koneksi ini berulang kali, beri tahu vendor proxy penyimpanan kunci eksternal Anda.

`XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION`

Konfigurasi layanan titik akhir Amazon VPC tidak sesuai dengan persyaratan untuk penyimpanan kunci eksternal AWS KMS .

- Layanan titik akhir VPC harus berupa layanan titik akhir untuk titik akhir antarmuka di pemanggil. Akun AWS
- Ini harus memiliki penyeimbang beban jaringan (NLB) yang terhubung ke setidaknya dua subnet, masing-masing di Availability Zone yang berbeda.
- `Allow principals`Daftar harus mencakup kepala AWS KMS layanan untuk Wilayah`cks.kms.<region>.amazonaws.com`, seperti`cks.kms.us-east-1.amazonaws.com`.
- Itu tidak boleh memerlukan penerimaan permintaan koneksi.
- Itu harus memiliki nama DNS pribadi. Nama DNS pribadi untuk penyimpanan kunci eksternal dengan `VPC_ENDPOINT_SERVICE` konektivitas harus unik di dalamnya Wilayah AWS.
- Domain nama DNS pribadi harus memiliki status verifikasi. `verified`
- Sertifikat TLS menentukan nama host DNS pribadi di mana titik akhir dapat dijangkau.

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

AWS KMS tidak dapat menemukan layanan titik akhir VPC yang digunakannya untuk berkomunikasi dengan proxy penyimpanan kunci eksternal. Verifikasi bahwa `XksProxyVpcEndpointServiceName` itu benar dan kepala AWS KMS layanan memiliki izin konsumen layanan di layanan titik akhir Amazon VPC.

Kesalahan latensi dan batas waktu

Pengecualian: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (operasi kriptografi), `XksProxyUriUnreachableException` (operasi manajemen)

Kode kesalahan koneksi:`XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Ketika tidak AWS KMS dapat menghubungi proxy dalam interval batas waktu 250 milidetik, ia mengembalikan pengecualian. `CreateCustomKeyStore`dan `UpdateCustomKeyStore` kembali`XksProxyUriUnreachableException`. Operasi kriptografi mengembalikan standar `KMSInvalidStateException` dengan pesan kesalahan yang menjelaskan masalah. Jika `ConnectCustomKeyStore` gagal, AWS KMS mengembalikan kode kesalahan koneksi yang menjelaskan masalah.

Kesalahan batas waktu mungkin merupakan masalah sementara yang dapat diselesaikan dengan mencoba kembali permintaan. Jika masalah berlanjut, verifikasi bahwa proxy penyimpanan kunci

eksternal Anda aktif dan terhubung ke jaringan, dan bahwa titik akhir URI proxy, jalur URI proxy, dan nama layanan titik akhir VPC (jika ada) sudah benar di penyimpanan kunci eksternal Anda. Juga, verifikasi bahwa manajer kunci eksternal Anda dekat dengan Wilayah AWS untuk penyimpanan kunci eksternal Anda. Jika Anda perlu memperbarui salah satu dari nilai-nilai ini, lihatMengedit properti penyimpanan kunci eksternal.

Untuk melacak pola latensi, gunakan `XksProxyLatency` CloudWatch metrik dan grafik latensi rata-rata (berdasarkan metrik tersebut) di bagian AWS KMS Pemantauan konsol. Proxy penyimpanan kunci eksternal Anda mungkin juga menghasilkan log dan metrik yang melacak latensi dan batas waktu.

`XksProxyUriUnreachableException`
AWS KMS tidak dapat berkomunikasi dengan proxy penyimpanan kunci eksternal. Ini mungkin masalah jaringan sementara. Jika Anda melihat kesalahan ini berulang kali, verifikasi bahwa proxy penyimpanan kunci eksternal Anda aktif dan terhubung ke jaringan, dan URI titik akhir sudah benar di penyimpanan kunci eksternal Anda.

- Proxy penyimpanan kunci eksternal tidak menanggapi permintaan API AWS KMS proxy dalam interval batas waktu 250 milidetik. Ini mungkin menunjukkan masalah jaringan sementara atau masalah operasional atau kinerja dengan proxy. Jika mencoba lagi tidak menyelesaikan masalah, beri tahu administrator proxy penyimpanan kunci eksternal Anda.

Kesalahan latensi dan batas waktu sering bermanifestasi sebagai kegagalan koneksi. Ketika ConnectCustomKeyStoreoperasi gagal, status koneksi penyimpanan kunci eksternal berubah `FAILED` dan AWS KMS mengembalikan kode kesalahan koneksi yang menjelaskan kesalahan. Untuk daftar kode kesalahan koneksi dan saran untuk menyelesaikan kesalahan, lihatKode kesalahan koneksi untuk penyimpanan kunci eksternal. Daftar kode koneksi untuk Semua toko kunci khusus dan toko kunci eksternal berlaku untuk toko kunci eksternal. Kesalahan koneksi berikut terkait dengan latensi dan batas waktu.

`XKS_PROXY_NOT_REACHABLE`

-atau-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`

AWS KMS tidak dapat berkomunikasi dengan proxy penyimpanan kunci eksternal. Verifikasi bahwa proxy penyimpanan kunci eksternal Anda aktif dan terhubung ke jaringan, dan jalur URI dan titik akhir URI atau nama layanan VPC sudah benar di penyimpanan kunci eksternal Anda.

Kesalahan ini dapat terjadi karena alasan berikut:

- Proxy penyimpanan kunci eksternal tidak aktif dan atau tidak terhubung ke jaringan.
- Ada kesalahan pada titik akhir URI proxy, jalur URI proxy, atau nilai nama layanan titik akhir VPC (jika ada) dalam konfigurasi penyimpanan kunci eksternal. Untuk melihat konfigurasi penyimpanan kunci eksternal, gunakan DescribeCustomKeyStoresoperasi atau lihat halaman detail untuk penyimpanan kunci eksternal di AWS KMS konsol.
- Mungkin ada kesalahan konfigurasi jaringan, seperti kesalahan port, pada jalur jaringan antara AWS KMS dan proxy penyimpanan kunci eksternal. AWS KMS berkomunikasi dengan proxy penyimpanan kunci eksternal pada port 443. Nilai ini tidak dapat dikonfigurasi.
- Ketika proxy penyimpanan kunci eksternal melaporkan (sebagai GetHealthStatustanggapan) bahwa semua instance pengelola kunci eksternal beradaUNAVAILABLE, ConnectCustomKeyStoreoperasi gagal dengan a`ConnectionErrorCode`. `XKS_PROXY_NOT_REACHABLE` Untuk bantuan, lihat dokumentasi pengelola kunci eksternal Anda.
- Kesalahan ini dapat dihasilkan dari jarak fisik yang jauh antara manajer kunci eksternal dan Wilayah AWS dengan penyimpanan kunci eksternal. Latensi ping (network round-trip time (RTT)) antara manajer kunci eksternal Wilayah AWS dan eksternal tidak boleh lebih dari 35 milidetik. Anda mungkin harus membuat penyimpanan kunci eksternal di Wilayah AWS yang lebih dekat dengan manajer kunci eksternal, atau memindahkan manajer kunci eksternal ke pusat data yang lebih dekat ke Wilayah AWS.

`XKS_PROXY_TIMED_OUT`

-atau-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
AWS KMS menolak permintaan karena proxy penyimpanan kunci eksternal tidak merespons tepat waktu. Coba lagi permintaannya. Jika Anda melihat kesalahan ini berulang kali, laporkan ke administrator proxy penyimpanan kunci eksternal Anda.

Kesalahan ini dapat terjadi karena alasan berikut:

- Kesalahan ini dapat dihasilkan dari jarak fisik yang jauh antara manajer kunci eksternal dan proxy penyimpanan kunci eksternal. Jika memungkinkan, pindahkan proxy penyimpanan kunci eksternal lebih dekat ke manajer kunci eksternal.
- Kesalahan batas waktu dapat terjadi ketika proxy tidak dirancang untuk menangani volume dan frekuensi permintaan dari AWS KMS. Jika CloudWatch metrik Anda menunjukkan masalah terus-menerus, beri tahu administrator proxy penyimpanan kunci eksternal Anda.
- Kesalahan batas waktu dapat terjadi ketika koneksi antara pengelola kunci eksternal dan VPC Amazon untuk penyimpanan kunci eksternal tidak beroperasi dengan benar. Jika Anda menggunakan AWS Direct Connect, verifikasi bahwa VPC dan pengelola kunci eksternal Anda dapat berkomunikasi secara efektif. Untuk bantuan menyelesaikan masalah apa pun, lihat Pemecahan Masalah AWS Direct Connect di Panduan Pengguna. AWS Direct Connect

`XKS_PROXY_TIMED_OUT`

-atau-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
Proxy penyimpanan kunci eksternal tidak menanggapi permintaan dalam waktu yang ditentukan. Coba lagi permintaannya. Jika Anda melihat kesalahan ini berulang kali, laporkan ke administrator proxy penyimpanan kunci eksternal Anda.

- Kesalahan ini dapat dihasilkan dari jarak fisik yang jauh antara manajer kunci eksternal dan proxy penyimpanan kunci eksternal. Jika memungkinkan, pindahkan proxy penyimpanan kunci eksternal lebih dekat ke manajer kunci eksternal.

Kesalahan kredensi otentikasi

Pengecualian: `CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (operasi kriptografi), `XksProxyIncorrectAuthenticationCredentialException` (operasi manajemen selain) `CreateKey`

Anda membuat dan memelihara kredensi otentikasi untuk AWS KMS proxy penyimpanan kunci eksternal Anda. Kemudian Anda memberi tahu AWS KMS nilai kredensialnya saat Anda membuat penyimpanan kunci eksternal. Untuk mengubah kredensi otentikasi, lakukan perubahan pada proxy penyimpanan kunci eksternal Anda. Kemudian perbarui kredensi untuk toko kunci eksternal Anda. Jika proxy Anda memutar kredensi, Anda harus memperbarui kredensi untuk penyimpanan kunci eksternal Anda.

Jika proxy penyimpanan kunci eksternal tidak akan mengautentikasi permintaan yang ditandatangani dengan kredensi otentikasi proxy untuk penyimpanan kunci eksternal Anda, efeknya bergantung pada permintaan:

- `CreateCustomKeyStore`dan `UpdateCustomKeyStore` gagal dengan sebuah`XksProxyIncorrectAuthenticationCredentialException`.
- `ConnectCustomKeyStore`berhasil, tetapi koneksi gagal. Status koneksi adalah `FAILED` dan kode kesalahan koneksi adalah`INVALID_CREDENTIALS`. Lihat perinciannya di Kesalahan koneksi penyimpanan kunci eksternal.
- Operasi kriptografi kembali `KMSInvalidStateException` untuk semua kesalahan konfigurasi eksternal dan kesalahan status koneksi di penyimpanan kunci eksternal. Pesan kesalahan yang menyertainya menjelaskan masalah.

Proxy penyimpanan kunci eksternal menolak permintaan karena tidak dapat mengautentikasi. AWS KMS Verifikasi kredensyal untuk penyimpanan kunci eksternal Anda dan perbarui jika perlu.

Kesalahan ini dapat terjadi karena alasan berikut:

- ID kunci akses atau kunci akses rahasia untuk penyimpanan kunci eksternal tidak cocok dengan nilai yang ditetapkan pada proxy penyimpanan kunci eksternal.

  Untuk memperbaiki kesalahan ini, perbarui kredensi otentikasi proxy untuk penyimpanan kunci eksternal Anda. Anda dapat membuat perubahan ini tanpa memutuskan penyimpanan kunci eksternal Anda.
- Proxy terbalik antara AWS KMS dan proxy penyimpanan kunci eksternal dapat memanipulasi header HTTP dengan cara yang membatalkan tanda tangan SigV4. Untuk memperbaiki kesalahan ini, beri tahu administrator proxy Anda.

Kesalahan status kunci

Pengecualian: `KMSInvalidStateException`

`KMSInvalidStateException`digunakan untuk dua tujuan berbeda untuk kunci KMS di toko kunci kustom.

- Ketika operasi manajemen, seperti`CancelKeyDeletion`, gagal dan mengembalikan pengecualian ini, ini menunjukkan bahwa status kunci dari kunci KMS tidak kompatibel dengan operasi.
- Ketika operasi kriptografi pada kunci KMS di toko kunci kustom gagal`KMSInvalidStateException`, itu dapat menunjukkan masalah dengan status kunci dari kunci KMS. Tetapi operasi AWS KMS kriptografi kembali `KMSInvalidStateException` untuk semua kesalahan konfigurasi eksternal dan kesalahan status koneksi di penyimpanan kunci eksternal. Untuk mengidentifikasi masalah, gunakan pesan kesalahan yang menyertai pengecualian.

Untuk menemukan status kunci yang diperlukan untuk operasi AWS KMS API, lihatStatus AWS KMS kunci kunci. Untuk menemukan status kunci kunci KMS, pada halaman Kunci yang dikelola Pelanggan, lihat bidang Status kunci KMS. Atau, gunakan DescribeKeyoperasi dan lihat `KeyState` elemen dalam respons. Lihat perinciannya di Melihat kunci.

> **Note**
>
> Status kunci dari kunci KMS di penyimpanan kunci eksternal tidak menunjukkan apa pun tentang status kunci eksternal yang terkait. Untuk informasi tentang status kunci eksternal, gunakan pengelola kunci eksternal dan alat proxy penyimpanan kunci eksternal. `CustomKeyStoreInvalidStateException`Ini mengacu pada keadaan koneksi penyimpanan kunci eksternal, bukan status kunci dari kunci KMS.

Operasi kriptografi pada kunci KMS di toko kustom mungkin gagal karena status kunci dari kunci KMS adalah atau. `Unavailable` `PendingDeletion` (Kunci dinonaktifkan kembali`DisabledException`.)

- Kunci KMS memiliki status `Disabled` kunci hanya ketika Anda sengaja menonaktifkan kunci KMS di AWS KMS konsol atau dengan menggunakan operasi. DisableKey Sementara kunci KMS dinonaktifkan, Anda dapat melihat dan mengelola kunci, tetapi Anda tidak dapat menggunakannya

dalam operasi kriptografi. Untuk memperbaiki masalah ini, aktifkan kuncinya. Lihat perinciannya di Mengaktifkan dan menonaktifkan kunci.

- Kunci KMS memiliki status `Unavailable` kunci ketika penyimpanan kunci eksternal terputus dari proxy penyimpanan kunci eksternal. Untuk memperbaiki kunci KMS yang tidak tersedia, sambungkan kembali penyimpanan kunci eksternal. Setelah penyimpanan kunci eksternal terhubung kembali, status kunci kunci KMS di penyimpanan kunci eksternal secara otomatis dikembalikan ke keadaan sebelumnya, seperti `Enabled` atau. `Disabled`

  Kunci KMS memiliki status `PendingDeletion` kunci ketika telah dijadwalkan untuk dihapus dan sedang dalam masa tunggu. Kesalahan status kunci pada kunci KMS yang tertunda penghapusan menunjukkan bahwa kunci tidak boleh dihapus, baik karena sedang digunakan untuk enkripsi, atau diperlukan untuk dekripsi. Untuk mengaktifkan kembali kunci KMS, batalkan penghapusan terjadwal, dan kemudian aktifkan kunci. Lihat perinciannya di Menjadwalkan dan membatalkan penghapusan kunci.

Kesalahan dekripsi

Pengecualian: `KMSInvalidStateException`

Ketika operasi Dekripsi dengan kunci KMS di penyimpanan kunci eksternal gagal, AWS KMS mengembalikan standar `KMSInvalidStateException` yang digunakan operasi kriptografi untuk semua kesalahan konfigurasi eksternal dan kesalahan status koneksi pada penyimpanan kunci eksternal. Pesan kesalahan menunjukkan masalah.

Untuk mendekripsi ciphertext yang dienkripsi menggunakan enkripsi ganda, manajer kunci eksternal pertama menggunakan kunci eksternal untuk mendekripsi lapisan luar ciphertext. Kemudian AWS KMS gunakan bahan AWS KMS kunci dalam kunci KMS untuk mendekripsi lapisan dalam ciphertext. Ciphertext yang tidak valid atau rusak dapat ditolak oleh manajer kunci eksternal atau. AWS KMS

Pesan kesalahan berikut menyertai `KMSInvalidStateException` ketika dekripsi gagal. Ini menunjukkan masalah dengan ciphertext atau konteks enkripsi opsional dalam permintaan.

> Proxy penyimpanan kunci eksternal menolak permintaan karena ciphertext yang ditentukan atau data tambahan yang diautentikasi rusak, hilang, atau tidak valid.

- Ketika proxy penyimpanan kunci eksternal atau manajer kunci eksternal melaporkan bahwa ciphertext atau konteks enkripsi tidak valid, biasanya menunjukkan masalah dengan ciphertext

atau konteks enkripsi dalam permintaan yang dikirim ke. `Decrypt` AWS KMS Untuk `Decrypt` operasi, AWS KMS kirimkan proxy ciphertext dan konteks enkripsi yang sama yang diterimanya dalam permintaan. `Decrypt`

Kesalahan ini mungkin disebabkan oleh masalah jaringan dalam perjalanan, seperti bit terbalik. Coba lagi `Decrypt` permintaannya. Jika masalah berlanjut, verifikasi bahwa ciphertext tidak diubah atau rusak. Juga, verifikasi bahwa konteks enkripsi dalam `Decrypt` permintaan untuk AWS KMS mencocokkan konteks enkripsi dalam permintaan yang mengenkripsi data.

Ciphertext yang dikirimkan oleh proxy penyimpanan kunci eksternal untuk dekripsi, atau konteks enkripsi, rusak, hilang, atau tidak valid.

- Ketika AWS KMS menolak ciphertext yang diterima dari proxy, ini menunjukkan bahwa manajer kunci eksternal atau proxy mengembalikan ciphertext yang tidak valid atau rusak ke. AWS KMS

  Kesalahan ini mungkin disebabkan oleh masalah jaringan dalam perjalanan, seperti bit terbalik. Coba lagi `Decrypt` permintaannya. Jika masalah berlanjut, verifikasi bahwa pengelola kunci eksternal beroperasi dengan benar, dan bahwa proxy penyimpanan kunci eksternal tidak mengubah ciphertext yang diterimanya dari pengelola kunci eksternal sebelum mengembalikannya. AWS KMS

Kesalahan kunci eksternal

Kunci eksternal adalah kunci kriptografi di manajer kunci eksternal yang berfungsi sebagai bahan kunci eksternal untuk kunci KMS. AWS KMS tidak dapat langsung mengakses kunci eksternal. Ini harus meminta manajer kunci eksternal (melalui proxy penyimpanan kunci eksternal) untuk menggunakan kunci eksternal untuk mengenkripsi data atau mendekripsi ciphertext.

Anda menentukan ID kunci eksternal di pengelola kunci eksternal ketika Anda membuat kunci KMS di penyimpanan kunci eksternal Anda. Anda tidak dapat mengubah ID kunci eksternal setelah kunci KMS dibuat. Untuk mencegah masalah dengan kunci KMS, `CreateKey` operasi meminta proxy penyimpanan kunci eksternal untuk memverifikasi ID dan konfigurasi kunci eksternal. Jika kunci eksternal tidak memenuhi persyaratan untuk digunakan dengan kunci KMS, `CreateKey` operasi gagal dengan pengecualian dan pesan kesalahan yang mengidentifikasi masalah.

Namun, masalah dapat terjadi setelah kunci KMS dibuat. Jika operasi kriptografi gagal karena masalah dengan kunci eksternal, operasi gagal dan mengembalikan pesan kesalahan yang menunjukkan masalah. `KMSInvalidStateException`

CreateKey kesalahan untuk kunci eksternal

Pengecualian:`XksKeyAlreadyInUseException`,, `XksKeyNotFoundException` `XksKeyInvalidConfigurationException`

CreateKeyOperasi mencoba memverifikasi ID dan properti kunci eksternal yang Anda berikan di parameter ID kunci eksternal (konsol) atau `XksKeyId` (API). Praktik ini dirancang untuk mendeteksi kesalahan lebih awal sebelum Anda mencoba menggunakan kunci eksternal dengan kunci KMS.

Kunci eksternal digunakan

Setiap kunci KMS di toko kunci eksternal harus menggunakan kunci eksternal yang berbeda. Ketika `CreateKey` mengenali bahwa ID kunci eksternal (XksKeyId) untuk kunci KMS tidak unik di penyimpanan kunci eksternal, gagal dengan file. `XksKeyAlreadyInUseException`

Jika Anda menggunakan beberapa ID untuk kunci eksternal yang sama, tidak `CreateKey` akan mengenali duplikat. Namun, kunci KMS dengan kunci eksternal yang sama tidak dapat dioperasikan karena memiliki bahan AWS KMS kunci dan metadata yang berbeda.

Kunci eksternal tidak ditemukan

Ketika proxy penyimpanan kunci eksternal melaporkan bahwa ia tidak dapat menemukan kunci eksternal menggunakan ID kunci eksternal (XksKeyId) untuk kunci KMS, `CreateKey` operasi gagal dan kembali `XksKeyNotFoundException` dengan pesan kesalahan berikut.

> Proxy penyimpanan kunci eksternal menolak permintaan karena tidak dapat menemukan kunci eksternal.

Kesalahan ini dapat terjadi karena alasan berikut:

- ID kunci eksternal (`XksKeyId`) untuk kunci KMS mungkin tidak valid. Untuk menemukan ID untuk proxy kunci eksternal yang digunakan untuk mengidentifikasi kunci eksternal, lihat proxy penyimpanan kunci eksternal atau dokumentasi pengelola kunci eksternal Anda.
- Kunci eksternal mungkin telah dihapus dari pengelola kunci eksternal Anda. Untuk menyelidiki, gunakan alat pengelola kunci eksternal Anda. Jika kunci eksternal dihapus secara permanen,

gunakan kunci eksternal yang berbeda dengan tombol KMS. Untuk daftar atau persyaratan untuk kunci eksternal, lihatPersyaratan untuk kunci KMS di toko kunci eksternal.

Persyaratan kunci eksternal tidak terpenuhi

Ketika proxy penyimpanan kunci eksternal melaporkan bahwa kunci eksternal tidak memenuhi persyaratan untuk digunakan dengan kunci KMS, `CreateKey` operasi gagal dan kembali `XksKeyInvalidConfigurationException` dengan salah satu pesan kesalahan berikut.

Spesifikasi kunci dari kunci eksternal harus AES_256. <key-spec>Spesifikasi kunci dari kunci eksternal yang ditentukan adalah.

- Kunci eksternal harus berupa kunci enkripsi simetris 256-bit dengan spesifikasi kunci AES_256. Jika kunci eksternal yang ditentukan adalah tipe yang berbeda, tentukan ID kunci eksternal yang memenuhi persyaratan ini.

Status kunci eksternal harus DIAKTIFKAN. Status kunci eksternal yang ditentukan adalah<status>.

- Kunci eksternal harus diaktifkan di manajer kunci eksternal. Jika kunci eksternal yang ditentukan tidak diaktifkan, gunakan alat pengelola kunci eksternal Anda untuk mengaktifkannya, atau tentukan kunci eksternal yang diaktifkan.

Penggunaan kunci dari kunci eksternal harus mencakup ENKRIPSI dan DEKRIPSI. Penggunaan kunci dari kunci eksternal yang ditentukan adalah < *key-usage* >.

- Kunci eksternal harus dikonfigurasi untuk enkripsi dan dekripsi di manajer kunci eksternal. Jika kunci eksternal yang ditentukan tidak menyertakan operasi ini, gunakan alat pengelola kunci eksternal Anda untuk mengubah operasi, atau tentukan kunci eksternal yang berbeda.

Kesalahan operasi kriptografi untuk kunci eksternal

Pengecualian: `KMSInvalidStateException`

Ketika proxy penyimpanan kunci eksternal tidak dapat menemukan kunci eksternal yang terkait dengan kunci KMS, atau kunci eksternal tidak memenuhi persyaratan untuk digunakan dengan kunci KMS, operasi kriptografi gagal.

Masalah kunci eksternal yang terdeteksi selama operasi kriptografi lebih sulit diselesaikan daripada masalah kunci eksternal yang terdeteksi sebelum membuat kunci KMS. Anda tidak dapat mengubah ID kunci eksternal setelah kunci KMS dibuat. Jika kunci KMS belum mengenkripsi data apa pun, Anda dapat menghapus kunci KMS dan membuat yang baru dengan ID kunci eksternal yang berbeda. Namun, ciphertext yang dihasilkan dengan kunci KMS tidak dapat didekripsi oleh kunci KMS lainnya, bahkan yang memiliki kunci eksternal yang sama, karena kunci akan memiliki metadata kunci yang berbeda dan bahan kunci yang berbeda. AWS KMS Sebagai gantinya, sejauh mungkin, gunakan alat pengelola kunci eksternal Anda untuk menyelesaikan masalah dengan kunci eksternal.

Ketika proxy penyimpanan kunci eksternal melaporkan masalah dengan kunci eksternal, operasi kriptografi kembali `KMSInvalidStateException` dengan pesan kesalahan yang mengidentifikasi masalah.

Kunci eksternal tidak ditemukan

Ketika proxy penyimpanan kunci eksternal melaporkan bahwa ia tidak dapat menemukan kunci eksternal menggunakan ID kunci eksternal (XksKeyId) untuk kunci KMS, operasi kriptografi mengembalikan a `KMSInvalidStateException` dengan pesan kesalahan berikut.

> Proxy penyimpanan kunci eksternal menolak permintaan karena tidak dapat menemukan kunci eksternal.

Kesalahan ini dapat terjadi karena alasan berikut:

- ID kunci eksternal (`XksKeyId`) untuk kunci KMS tidak lagi valid.

  Untuk menemukan ID kunci eksternal yang terkait dengan kunci KMS Anda, lihat detail kunci KMS. Untuk menemukan ID yang digunakan proxy kunci eksternal untuk mengidentifikasi kunci eksternal, lihat proxy penyimpanan kunci eksternal atau dokumentasi pengelola kunci eksternal.

  AWS KMS memverifikasi ID kunci eksternal saat membuat kunci KMS di toko kunci eksternal. Namun, ID mungkin menjadi tidak valid, terutama jika nilai ID kunci eksternal adalah alias atau nama yang bisa berubah. Anda tidak dapat mengubah ID kunci eksternal yang terkait dengan kunci

KMS yang ada. Untuk mendekripsi ciphertext apa pun yang dienkripsi di bawah kunci KMS, Anda harus mengaitkan kembali kunci eksternal dengan ID kunci eksternal yang ada.

Jika Anda belum menggunakan kunci KMS untuk mengenkripsi data, Anda dapat membuat kunci KMS baru dengan ID kunci eksternal yang valid. Namun, jika Anda telah membuat ciphertext dengan kunci KMS, Anda tidak dapat menggunakan kunci KMS lain untuk mendekripsi ciphertext, bahkan jika menggunakan kunci eksternal yang sama.

- Kunci eksternal mungkin telah dihapus dari pengelola kunci eksternal Anda. Untuk menyelidiki, gunakan alat pengelola kunci eksternal Anda. Jika memungkinkan, cobalah untuk memulihkan materi kunci dari salinan atau cadangan manajer kunci eksternal Anda. Jika kunci eksternal dihapus secara permanen, setiap ciphertext yang dienkripsi di bawah kunci KMS terkait tidak dapat dipulihkan.

Kesalahan konfigurasi kunci eksternal

Ketika proxy penyimpanan kunci eksternal melaporkan bahwa kunci eksternal tidak memenuhi persyaratan untuk digunakan dengan kunci KMS, operasi kriptografi kembali `KMSInvalidStateException` dengan salah satu pesan kesalahan berikut.

Proxy penyimpanan kunci eksternal menolak permintaan karena kunci eksternal tidak mendukung operasi yang diminta.

- Kunci eksternal harus mendukung enkripsi dan dekripsi. Jika penggunaan kunci tidak termasuk enkripsi dan dekripsi, gunakan alat pengelola kunci eksternal Anda untuk mengubah penggunaan kunci.

Proxy penyimpanan kunci eksternal menolak permintaan karena kunci eksternal tidak diaktifkan di pengelola kunci eksternal.

- Kunci eksternal harus diaktifkan dan tersedia untuk digunakan di pengelola kunci eksternal. Jika status kunci eksternal tidak`Enabled`, gunakan alat pengelola kunci eksternal Anda untuk mengaktifkannya.

Masalah proxy

Pengecualian:

`CustomKeyStoreInvalidStateException`(`CreateKey`), `KMSInvalidStateException` (operasi kriptografi),`UnsupportedOperationException`,`XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (operasi manajemen selain`CreateKey`)

Proxy penyimpanan kunci eksternal memediasi semua komunikasi antara AWS KMS dan manajer kunci eksternal. Ini menerjemahkan AWS KMS permintaan generik ke dalam format yang dapat dipahami oleh manajer kunci eksternal Anda. Jika proxy penyimpanan kunci eksternal tidak sesuai dengan Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal, atau jika tidak beroperasi dengan benar, atau tidak dapat berkomunikasi AWS KMS, Anda tidak akan dapat membuat atau menggunakan kunci KMS di penyimpanan kunci eksternal Anda.

Sementara banyak kesalahan menyebutkan proxy penyimpanan kunci eksternal karena peran pentingnya dalam arsitektur penyimpanan kunci eksternal, masalah tersebut mungkin berasal dari manajer kunci eksternal atau kunci eksternal.

Masalah di bagian ini berhubungan dengan masalah dengan desain atau pengoperasian proxy penyimpanan kunci eksternal. Menyelesaikan masalah ini mungkin memerlukan perubahan pada perangkat lunak proxy. Konsultasikan dengan administrator proxy Anda. Untuk membantu mendiagnosis masalah proxy, AWS KMS berikan XKS Proxy Text Client, klien pengujian open source yang memverifikasi bahwa proxy penyimpanan kunci eksternal Anda sesuai dengan Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` atau `XksProxyUriUnreachableException`
Proxy penyimpanan kunci eksternal dalam keadaan tidak sehat. Jika Anda melihat pesan ini berulang kali, beri tahu administrator proxy penyimpanan kunci eksternal Anda.

- Kesalahan ini dapat menunjukkan masalah operasional atau kesalahan perangkat lunak di proxy penyimpanan kunci eksternal. Anda dapat menemukan entri CloudTrail log untuk operasi AWS KMS API yang menghasilkan setiap kesalahan. Kesalahan ini dapat diatasi dengan mencoba kembali operasi. Namun, jika tetap ada, beri tahu administrator proxy penyimpanan kunci eksternal Anda.

- Ketika proxy penyimpanan kunci eksternal melaporkan (sebagai GetHealthStatustanggapan) bahwa semua instance pengelola kunci eksternal beradaUNAVAILABLE, upaya untuk membuat atau memperbarui penyimpanan kunci eksternal gagal dengan pengecualian ini. Jika kesalahan ini berlanjut, lihat dokumentasi pengelola kunci eksternal Anda.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` atau `XksProxyInvalidResponseException`

AWS KMS tidak dapat menafsirkan respons dari proxy penyimpanan kunci eksternal. Jika Anda melihat kesalahan ini berulang kali, konsultasikan dengan administrator proxy penyimpanan kunci eksternal Anda.

- AWS KMS operasi menghasilkan pengecualian ini ketika proxy mengembalikan respons yang tidak ditentukan yang tidak AWS KMS dapat mengurai atau menafsirkan. Kesalahan ini mungkin terjadi sesekali karena masalah eksternal sementara atau kesalahan jaringan sporadis. Namun, jika tetap ada, ini mungkin menunjukkan bahwa proxy penyimpanan kunci eksternal tidak sesuai dengan Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal. Beri tahu administrator atau vendor toko kunci eksternal Anda.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` atau `UnsupportedOperationException`

Proxy penyimpanan kunci eksternal menolak permintaan karena tidak mendukung operasi kriptografi yang diminta.

- Proxy penyimpanan kunci eksternal harus mendukung semua API proxy yang ditentukan dalam Spesifikasi API Proxy Toko Kunci AWS KMS Eksternal. Kesalahan ini menunjukkan bahwa proxy tidak mendukung operasi yang terkait dengan permintaan. Beri tahu administrator atau vendor toko kunci eksternal Anda.

Masalah otorisasi proxy

Pengecualian:`CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Beberapa proxy penyimpanan kunci eksternal menerapkan persyaratan otorisasi untuk penggunaan kunci eksternalnya. Proxy penyimpanan kunci eksternal diizinkan, tetapi tidak diperlukan, untuk merancang dan mengimplementasikan skema otorisasi yang memungkinkan pengguna tertentu untuk meminta operasi tertentu dalam kondisi tertentu. Misalnya, proxy mungkin memungkinkan pengguna untuk mengenkripsi dengan kunci eksternal tertentu, tetapi tidak untuk mendekripsi dengannya. Untuk informasi selengkapnya, lihat Otorisasi proxy penyimpanan kunci eksternal (opsional).

Otorisasi proxy didasarkan pada metadata yang AWS KMS termasuk dalam permintaannya ke proxy. `awsSourceVpce`Bidang `awsSourceVpc` dan disertakan dalam metadata hanya jika permintaan berasal dari titik akhir VPC dan hanya ketika pemanggil berada di akun yang sama dengan kunci KMS.

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

Ketika proxy menolak permintaan karena kegagalan otorisasi, AWS KMS operasi terkait gagal. `CreateKey`kembali`CustomKeyStoreInvalidStateException`. AWS KMS operasi kriptografi kembali`KMSInvalidStateException`. Keduanya menggunakan pesan kesalahan berikut:

> Proxy penyimpanan kunci eksternal menolak akses ke operasi. Verifikasi bahwa pengguna dan kunci eksternal keduanya diotorisasi untuk operasi ini, dan coba permintaan lagi.

- Untuk mengatasi kesalahan, gunakan pengelola kunci eksternal atau alat proxy penyimpanan kunci eksternal untuk menentukan mengapa otorisasi gagal. Kemudian, perbarui prosedur yang menyebabkan permintaan tidak sah atau gunakan alat proxy penyimpanan kunci eksternal Anda untuk memperbarui kebijakan otorisasi. Anda tidak dapat menyelesaikan kesalahan ini di AWS KMS.

Referensi tipe kunci

AWS KMSmendukung fitur yang berbeda untuk berbagai jenis tombol KMS. Misalnya, Anda hanya dapat menggunakan kunci KMS enkripsi simetris untuk menghasilkan kunci data simetris dan pasangan kunci data asimetris. Selain itu, mengimpor bahan kunci dan rotasi kunci otomatis hanya didukung untuk kunci KMS enkripsi simetris, dan Anda hanya dapat membuat kunci KMS enkripsi simetris di toko kunci khusus.

Referensi ini mencakup dua tabel.

- Tabel tipe Kunci mencantumkan AWS KMS operasi yang valid untuk kunci KMS enkripsi simetris, kunci KMS asimetris, dan kunci KMS HMAC.
- Tabel Fitur Khusus mencantumkan AWS KMS operasi yang berlaku untuk kunci KMS Multi-wilayah, kunci KMS dengan bahan kunci yang diimpor, dan kunci KMS di toko kunci khusus.

Tabel tipe kunci

Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
CancelKeyDeletion	✓	✓	✓	✓
CreateAlias	✓	✓	✓	✓
CreateGrant	✓	✓	✓	✓
CreateKey	✓	✓	✓	✓
Dekripsi	✓	⊗	✓	⊗

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
DeleteAlias	✓	✓	✓	✓
DeleteImportedKeyMaterial Hanya berlaku pada kunci KMS dengan bahan kunci impor (`Origin`is`EXTERNAL`).	✓	✓	✓	✓
DescribeKey	✓	✓	✓	✓
DisableKey	✓	✓	✓	✓
DisableKeyRotation	✓ Hanya berlaku pada kunci KMS dengan materi AWS KMS kunci (`Origin`is`AW`	⊗	⊗	⊗
EnableKey	✓	✓	✓	✓

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
EnableKeyRotation	✓ Hanya berlaku pada kunci KMS dengan materi AWS KMS kunci (`OriginisAW`	⊗	⊗	⊗
Enkripsi	✓	⊗	✓	⊗
GenerateDataKey	✓	⊗	⊗	⊗
GenerateDataKeyPair Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	✓ Tidak berlaku pada kunci KMS di toko kunci khusus.	⊗	⊗	⊗

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
GenerateDataKeyPairWithoutPlaintext Menghasilkan data key pair asimetris yang dilindungi oleh kunci KMS enkripsi simetris.	✓ Tidak berlaku pada kunci KMS di toko kunci khusus.	⊗	⊗	⊗
GenerateDataKeyWithoutPlaintext	✓	⊗	⊗	⊗
GenerateMac	⊗	✓	⊗	⊗
GetKeyPolicy	✓	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` akan selalu berupa `false`.)	✓ (`KeyRotationEnabled` akan selalu berupa `false`.)	✓ (`KeyRotationEnabled` akan selalu berupa `false`.)
GetParametersForImport Hanya berlaku pada kunci KMS dengan bahan kunci impor (`Origin` is `EXTERNAL`).	✓	✓	✓	✓

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
GetPublicKey	⊗	⊗	✓	✓
ImportKeyMaterial Hanya berlaku pada kunci KMS dengan bahan kunci impor (`Origin`is`EXTERNAL`).	✓	✓	✓	✓
ListAliases	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓
ListResourceTags	✓	✓	✓	✓
ListRetirableGrants	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓
ReEncrypt	✓	⊗	✓	⊗
ReplicateKey - Hanya berlaku pada kunci Multi-wilayah	✓	✓	✓	✓
RetireGrant	✓	✓	✓	✓

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
RevokeGrant	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓	✓
Tanda	⊗	⊗	⊗	✓
TagResource	✓	✓	✓	✓
UntagResource	✓	✓	✓	✓
UpdateAlias Kunci KMS saat ini dan kunci KMS baru harus tipe yang sama (keduanya simetris atau keduanya asimetris atau keduanya HMAC) dan keduanya harus memiliki penggunaan kunci yang sama.	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✓	✓
UpdateReplicaRegion - Hanya berlaku pada kunci Multi-wilayah	✓	✓	✓	✓
Verifikasi	⊗	⊗	⊗	✓

Operasi API AWS KMS	Kunci KMS enkripsi simetris	Kunci HMAC KMS	Kunci KMS asimetris (ENCRYPT_DECRYPT)	Tombol KMS asimetris (SIGN_VERIFY)
VerifyMac	⊗	✓	⊗	⊗

Tabel fitur khusus

Tabel ini menunjukkan operasi AWS KMS API yang didukung pada setiap jenis kunci tujuan khusus.

Saat membaca tabel ini, waspadai interaksi berikut:

- Tombol Multi-Region:
  - Kunci Multi-Region dapat berupa kunci KMS enkripsi simetris, kunci KMS asimetris, kunci KMS HMAC, dan kunci KMS dengan bahan kunci yang diimpor.
  - Anda tidak dapat membuat kunci multi-Wilayah di penyimpanan kunci kustom.
- Bahan kunci yang diimpor
  - Anda dapat mengimpor bahan kunci untuk kunci KMS enkripsi simetris, kunci KMS asimetris, dan kunci KMS HMAC.
  - Anda dapat membuat kunci Multi-wilayah dengan bahan kunci yang diimpor.
  - Anda tidak dapat membuat kunci dengan materi kunci yang diimpor di toko kunci khusus.
  - Rotasi tombol otomatis (`EnableKeyRotation`,`DisableKeyRotation`) tidak didukung untuk kunci KMS dengan bahan kunci impor.
- Toko kunci khusus
  - Toko kunci khusus hanya mendukung kunci KMS enkripsi simetris.
  - Operasi simetris pada pasangan kunci asimetris (`GenerateDataKeyPair`,`GenerateDataKeyPairWithoutPlaintext`) tidak didukung pada kunci KMS di toko kunci khusus.
  - Rotasi kunci otomatis (`EnableKeyRotation`,`DisableKeyRotation`) tidak didukung pada kunci KMS di toko kunci khusus.
  - Anda tidak dapat membuat kunci Multi-wilayah di toko kunci khusus.

Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Anda dapat menggunakan `CreateKey` untuk membuat kunci utama Multi-wilayah, kunci KMS dengan bahan kunci impor, atau kunci KMS di toko kunci kustom. Untuk membuat kunci replika Multi-wilayah, gunakan. `ReplicateKey`	✓	✓	✓
Dekripsi	✓ Hanya berlaku `KeyUsage` ketika `ENCRYPT_DECRYPT`	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ Hanya berlaku untuk kunci	✓	⊗

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
	dengan bahan kunci impor (`OriginisEXTERN`		
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓
DisableKeyRotation	✓ Hanya berlaku pada kunci enkripsi simetris dengan material AWS KMS kunci (`OriginisAWS_KM`	⊗	⊗
EnableKey	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓
EnableKeyRotation	✓ Hanya berlaku pada kunci enkripsi simetris dengan material AWS KMS kunci (`OriginisAWS_KM`	⊗	⊗

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
Enkripsi	✓ Hanya berlaku KeyUsage ketika ENCRYPT_DECRYPT	✓	✓
GenerateDataKey	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓
GenerateDataKeyPair	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	⊗
GenerateDataKeyPairWithoutPlaintext	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ Hanya berlaku pada kunci KMS enkripsi simetris	✓	✓

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
GenerateMac Hanya berlaku pada kunci HMAC KMS	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (`KeyRotationEnabled` akan selalu berupa `false`.)	⊗
GetParametersForImport	✓ Hanya berlaku untuk kunci dengan bahan kunci impor (`OriginisEXTERN`	✓	⊗
GetPublicKey Hanya berlaku untuk kunci KMS asimetris.	✓	✓	⊗

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
ImportKeyMaterial	✓ Hanya berlaku untuk kunci dengan bahan kunci impor (`OriginisEXTERN`	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Hanya berlaku `KeyUsage` ketika `ENCRYPT_DECRYPT`	✓	✓

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
ReplicateKey	✓ Hanya berlaku pada kunci utama Multi-wilayah.	✓ Hanya berlaku pada kunci utama Multi-wilayah.	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓
Tanda Berlaku hanya pada saat `KeyUsage` `SIGN_VERIFY` .	✓	✓	⊗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓
UpdateAlias - Kunci KMS saat ini dan kunci KMS baru harus tipe yang sama (simetris atau keduanya asimetris atau keduanya HMAC) dan keduanya harus memiliki penggunaan kunci yang sama.	✓	✓	✓

Operasi API AWS KMS	Kunci Multi-Wilayah	Materi kunci yang diimpor	Kunci KMS di toko kunci khusus
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ Hanya berlaku pada kunci Multi-wilayah.	⊗
Verifikasi Hanya berlaku bila `KeyUsage` ada`SIGN_VERIFY` .	✓	✓	⊗
VerifyMac Hanya berlaku pada kunci HMAC KMS	✓	✓	⊗

Keamanan AWS Key Management Service

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari organisasi yang paling sensitif terhadap keamanan.

Keamanan menjadi tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan dari cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami secara berkala sebagai bagian dari AWSProgram Kepatuhan. Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS Key Management Service (AWS KMS), lihat AWSLayanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Selain konfigurasi dan penggunaan AndaAWS KMS keys, Anda bertanggung jawab atas faktor-faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, dan hukum dan peraturan yang berlaku AWS KMS

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Key Management Service. Topik berikut menunjukkan kepada Anda cara mengonfigurasi AWS KMS untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Topik

Perlindungan data di AWS Key Management Service

AWS Key Management Service menyimpan dan melindungi kunci enkripsi Anda untuk membuatnya sangat tersedia sambil memberi Anda kontrol akses yang kuat dan fleksibel.

Topik

Melindungi bahan utama

Secara default, AWS KMS menghasilkan dan melindungi materi kunci kriptografi untuk kunci KMS. Selain itu, AWS KMS menawarkan opsi untuk materi utama yang dibuat dan dilindungi di luarAWS KMS. Untuk detail teknis tentang kunci KMS dan materi utama, lihat Detail AWS Key Management ServiceKriptografi.

Melindungi material utama yang dihasilkan AWS KMS

Saat Anda membuat kunci KMS, secara default, AWS KMS menghasilkan dan melindungi materi kriptografi untuk kunci KMS.

Untuk melindungi materi kunci untuk kunci KMS, AWS KMS bergantung pada armada terdistribusi FIPS 140-2 Security Level 3 — modul keamanan perangkat keras (HSM) yang divalidasi. Setiap AWS KMS HSM adalah alat perangkat keras mandiri khusus yang dirancang untuk menyediakan fungsi kriptografi khusus untuk memenuhi persyaratan keamanan dan skalabilitas. AWS KMS (HSM yang AWS KMS digunakan di Wilayah China disertifikasi oleh OSCCA dan mematuhi semua peraturan Tiongkok terkait, tetapi tidak divalidasi berdasarkan Program Validasi Modul Kriptografi FIPS 140-2.)

Bahan kunci untuk kunci KMS dienkripsi secara default ketika dihasilkan di HSM. Materi kunci didekripsi hanya dalam memori volatile HSM dan hanya untuk beberapa milidetik yang diperlukan untuk menggunakannya dalam operasi kriptografi. Setiap kali bahan utama tidak digunakan secara aktif, itu dienkripsi dalam HSM dan ditransfer ke penyimpanan persisten latensi rendah yang sangat tahan lama (99,999999999%) yang tetap terpisah dan terisolasi dari HSM. Materi kunci Plaintext tidak pernah meninggalkan batas keamanan HSM; itu tidak pernah ditulis ke disk atau bertahan di media penyimpanan apa pun. (Satu-satunya pengecualian adalah kunci publik dari key pair asimetris, yang bukan rahasia.)

AWSmenegaskan sebagai prinsip keamanan mendasar bahwa tidak ada interaksi manusia dengan materi kunci kriptografi teks biasa dari jenis apa pun. Layanan AWS Tidak ada mekanisme bagi siapa pun, termasuk Layanan AWS operator, untuk melihat, mengakses, atau mengekspor materi kunci teks biasa. Prinsip ini berlaku bahkan selama kegagalan bencana dan peristiwa pemulihan bencana. Materi kunci pelanggan Plaintext AWS KMS digunakan untuk operasi kriptografi dalam HSM yang divalidasi AWS KMS FIPS hanya sebagai tanggapan atas permintaan resmi yang dibuat untuk layanan oleh pelanggan atau delegasi mereka.

Untuk kunci yang dikelola pelanggan, kunci Akun AWS yang menciptakan kunci adalah pemilik kunci tunggal dan tidak dapat dipindahtangankan. Akun pemilik memiliki kontrol penuh dan eksklusif atas kebijakan otorisasi yang mengontrol akses ke kunci. UntukKunci yang dikelola AWS, Akun AWS memiliki kontrol penuh atas kebijakan IAM yang mengotorisasi permintaan ke. Layanan AWS

Melindungi material utama yang dihasilkan di luar AWS KMS

AWS KMSmemberikan alternatif untuk bahan utama yang dihasilkan diAWS KMS.

Toko kunci khusus, AWS KMS fitur opsional, memungkinkan Anda membuat kunci KMS yang didukung oleh materi utama yang dihasilkan dan digunakan di luar. AWS KMS Kunci KMS di toko-toko AWS CloudHSM utama didukung oleh kunci dalam modul keamanan AWS CloudHSM perangkat keras yang Anda kontrol. HSM ini disertifikasi di FIPS 140-2 Security Level 3. Kunci KMS di toko kunci eksternal didukung oleh kunci di manajer kunci eksternal yang Anda kontrol dan kelola di luarAWS, seperti HSM fisik di pusat data pribadi Anda.

Fitur opsional lainnya memungkinkan Anda mengimpor bahan kunci untuk kunci KMS. Untuk melindungi material kunci yang diimpor saat sedang dalam perjalananAWS KMS, Anda mengenkripsi materi kunci menggunakan kunci publik dari key pair RSA yang dihasilkan dalam HSM. AWS KMS Bahan kunci yang diimpor didekripsi dalam AWS KMS HSM dan dienkripsi ulang di bawah kunci simetris di HSM. Seperti semua materi utama, materi AWS KMS kunci impor plaintext tidak pernah meninggalkan HSM tidak terenkripsi. Namun, pelanggan yang menyediakan bahan utama bertanggung jawab atas penggunaan yang aman, daya tahan, dan pemeliharaan material utama di luarAWS KMS.

Enkripsi data

Data AWS KMS terdiri dari AWS KMS keysdan bahan kunci enkripsi yang mereka wakili. Material kunci ini ada dalam plaintext hanya dalam modul keamanan perangkat keras (HSM) AWS KMS dan hanya saat digunakan. Jika tidak, material kunci dienkripsi dan disimpan dalam penyimpanan tetap.

Materi kunci yang AWS KMS menghasilkan kunci KMS tidak pernah meninggalkan batas HSM yang tidak terenkripsi. AWS KMS Ini tidak diekspor atau ditransmisikan dalam operasi API AWS KMS apa pun. Pengecualiannya adalah untuk kunci Multi-wilayah, di mana AWS KMS menggunakan mekanisme replikasi Lintas wilayah untuk menyalin materi kunci untuk kunci Multi-wilayah dari HSM dalam satu Wilayah AWS ke HSM yang berbeda. Wilayah AWS Untuk detailnya, lihat Proses replikasi untuk kunci Multi-wilayah di Detail AWS Key Management Service Kriptografi.

Topik

- Enkripsi diam
- Enkripsi dalam bergerak

Enkripsi diam

AWS KMSmenghasilkan materi utama untuk AWS KMS keys FIPS 140-2 Security Level 3 — compliant hardware security modules (HSM). Satu-satunya pengecualian adalah Wilayah China, di mana HSM yang AWS KMS digunakan untuk menghasilkan kunci KMS mematuhi semua peraturan China terkait, tetapi tidak divalidasi berdasarkan Program Validasi Modul Kriptografi FIPS 140-2. Saat tidak digunakan, bahan kunci dienkripsi oleh kunci HSM dan ditulis ke penyimpanan yang tahan lama dan persisten. Materi kunci untuk kunci KMS dan kunci enkripsi yang melindungi materi kunci tidak pernah meninggalkan HSM dalam bentuk teks biasa.

Enkripsi dan pengelolaan bahan kunci untuk kunci KMS ditangani sepenuhnya oleh. AWS KMS

Untuk detail selengkapnya, lihat Bekerja dengan AWS KMS keys Rincian AWS Key Management Service Kriptografi.

Enkripsi dalam bergerak

Materi kunci yang AWS KMS dihasilkan untuk kunci KMS tidak pernah diekspor atau ditransmisikan dalam operasi AWS KMS API. AWS KMSmenggunakan pengidentifikasi kunci untuk mewakili kunci KMS dalam operasi API. Demikian pula, materi kunci untuk kunci KMS di toko kunci AWS KMS khusus tidak dapat diekspor dan tidak pernah ditransmisikan dalam AWS KMS atau AWS CloudHSM operasi API.

Namun, beberapa operasi API AWS KMS mengembalikan kunci data. Selain itu, pelanggan dapat menggunakan operasi API untuk mengimpor materi kunci untuk kunci KMS yang dipilih.

Semua panggilan AWS KMS API harus ditandatangani dan ditransmisikan menggunakan Transport Layer Security (TLS). AWS KMSmembutuhkan TLS 1.2 dan merekomendasikan TLS 1.3 di semua

wilayah. AWS KMSjuga mendukung TLS pasca-kuantum hibrida untuk titik akhir AWS KMS layanan di semua wilayah, kecuali Wilayah China. AWS KMStidak mendukung TLS pasca-kuantum hibrida untuk titik akhir FIPS di. AWS GovCloud (US) Panggilan ke AWS KMS juga memerlukan suite penyandian modern yang mendukung kerahasiaan penerusan sempurna, yang berarti bahwa gangguan rahasia apa pun, seperti kunci privat, tidak mengganggu kunci sesi juga.

Jika Anda memerlukan modul kriptografi yang divalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk menggunakan endpoint standar atau AWS KMS endpoint AWS KMS FIPS, klien harus mendukung TLS 1.2 atau yang lebih baru. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2. Untuk daftar titik akhir AWS KMS FIPS, lihat AWS Key Management Servicetitik akhir dan kuota di. Referensi Umum AWS

Komunikasi antara host layanan AWS KMS dan HSM dilindungi menggunakan Elliptic Curve Cryptography (ECC) dan Advanced Encryption Standard (AES) dalam skema enkripsi diautentikasi. Untuk detail selengkapnya, lihat Keamanan komunikasi internal dalam Detail Kriptografi AWS Key Management Service.

Privasi lalu lintas antar jaringan

AWS KMSmendukung satu AWS Management Console dan satu set operasi API yang memungkinkan Anda untuk membuat dan mengelola AWS KMS keys dan menggunakannya dalam operasi kriptografi.

AWS KMS mendukung dua opsi konektivitas jaringan dari jaringan privat Anda ke AWS.

- Koneksi VPN IPSec melalui internet
- AWS Direct Connect, yang menautkan jaringan internal Anda ke lokasi AWS Direct Connect melalui kabel serat optik Ethernet standar.

Semua panggilan AWS KMS API harus ditandatangani dan ditransmisikan menggunakan Transport Layer Security (TLS). Panggilan juga memerlukan suite penyandian modern yang mendukung kerahasiaan penerusan sempurna. Lalu lintas ke modul keamanan perangkat keras (HSM) yang menyimpan materi kunci untuk kunci KMS hanya diizinkan dari host AWS KMS API yang diketahui melalui jaringan AWS internal.

Untuk terhubung langsung ke AWS KMS dari virtual private cloud (VPC) Anda tanpa mengirimkan lalu lintas melalui internet publik, gunakan titik akhir VPC, yang didukung oleh AWS PrivateLink. Untuk informasi selengkapnya, lihat Terhubung ke AWS KMS melalui VPC endpoint.

AWS KMS juga mendukung pertukaran kunci pasca-kuantum hibrida untuk protokol enkripsi jaringan Transport Layer Security (TLS). Anda dapat menggunakan opsi ini dengan TLS ketika Anda terhubung ke titik akhir API AWS KMS.

Identity and access management untuk AWS Key Management Service

AWS Identity and Access Management(IAM) membantu Anda mengontrol akses ke AWS sumber daya dengan aman. Administrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS KMS Untuk informasi selengkapnya, lihat Menggunakan kebijakan IAM dengan AWS KMS.

Kebijakan kunci adalah mekanisme utama untuk mengontrol akses ke kunci KMS diAWS KMS. Setiap kunci KMS harus memiliki kebijakan kunci. Anda juga dapat menggunakan kebijakan dan hibah IAM, bersama dengan kebijakan utama, untuk mengontrol akses ke kunci KMS Anda. Untuk informasi selengkapnya, lihat Kontrol autentikasi dan akses untuk AWS KMS.

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC), Anda dapat membuat antarmuka VPC endpoint untuk didukung oleh. AWS KMS AWS PrivateLink Anda juga dapat menggunakan kebijakan titik akhir VPC untuk menentukan prinsipal mana yang dapat mengakses AWS KMS titik akhir Anda, panggilan API mana yang dapat mereka lakukan, dan kunci KMS mana yang dapat mereka akses. Lihat perinciannya di Mengontrol akses ke VPC endpoint.

Pencatatan dan pemantauan di AWS Key Management Service

Pemantauan adalah bagian penting untuk memahami ketersediaan, keadaan, dan penggunaan situs Anda AWS KMS keysAWS KMS. Pemantauan membantu menjaga keamanan, keandalan, ketersediaan, dan kinerja AWS solusi Anda. AWSmenyediakan beberapa alat untuk memantau kunci KMS Anda.

Log AWS CloudTrail

Setiap panggilan ke operasi AWS KMS API ditangkap sebagai peristiwa dalam AWS CloudTrail log. Log ini merekam semua panggilan API dari konsol AWS KMS, dan panggilan yang dibuat oleh AWS KMS dan llayanan AWS lain. Panggilan API lintas akun, seperti panggilan untuk menggunakan kunci KMS secara berbedaAkun AWS, dicatat dalam CloudTrail log kedua akun.

Saat memecahkan masalah atau mengaudit, Anda dapat menggunakan log untuk merekonstruksi siklus hidup kunci KMS. Anda juga dapat melihat manajemen dan penggunaan kunci KMS dalam

operasi kriptografi. Untuk informasi selengkapnya, lihat the section called “Logging dengan AWS CloudTrail”.

CloudWatch Log Amazon

Memantau, menyimpan, dan mengakses berkas log Anda dari AWS CloudTrail dan sumber lain. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

UntukAWS KMS, CloudWatch menyimpan informasi berguna yang membantu Anda mencegah masalah dengan kunci KMS Anda dan sumber daya yang mereka lindungi. Untuk informasi selengkapnya, lihat the section called “Pemantauan CloudWatch dengan”.

Amazon EventBridge

AWS KMSmenghasilkan EventBridge peristiwa ketika kunci KMS Anda diputar atau dihapus atau materi kunci yang diimpor dalam kunci KMS Anda kedaluwarsa. Cari peristiwa AWS KMS (operasi API) dan arahkan mereka ke satu atau lebih fungsi atau stream target untuk menangkap informasi status. Untuk informasi selengkapnya, lihat the section called “Pemantauan EventBridge dengan Amazon” dan Panduan EventBridge Pengguna Amazon.

CloudWatch Metrik Amazon

Anda dapat memantau kunci KMS menggunakan CloudWatch metrik, yang mengumpulkan dan memproses data mentah dari AWS KMS metrik kinerja. Data dicatat dalam interval dua minggu sehingga Anda dapat melihat tren informasi terkini dan historis. Ini membantu Anda memahami bagaimana kunci KMS Anda digunakan dan bagaimana penggunaannya berubah dari waktu ke waktu. Untuk informasi tentang penggunaan CloudWatch metrik untuk memantau kunci KMS, lihat. AWS KMS metrik dan dimensi

CloudWatch Alarm Amazon

Lihat satu metrik berubah selama suatu periode waktu yang Anda tentukan. Lalu lakukan tindakan berdasarkan nilai metrik relatif terhadap ambang batas selama sejumlah periode waktu. Misalnya, Anda dapat membuat CloudWatch alarm yang dipicu ketika seseorang mencoba menggunakan kunci KMS yang dijadwalkan akan dihapus dalam operasi kriptografi. Ini menunjukkan bahwa kunci KMS masih digunakan dan mungkin tidak boleh dihapus. Untuk informasi selengkapnya, lihat the section called “Membuat alarm”.

AWS Security Hub

Anda dapat memantau AWS KMS penggunaan Anda untuk standar industri keamanan dan praktik terbaik kepatuhan menggunakanAWS Security Hub. Hub Keamanan menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan guna membantu

Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya, lihat AWS Key Management Servicekontrol di Panduan AWS Security Hub Pengguna.

Validasi kepatuhan untuk AWS Key Management Service

Auditor pihak ketiga melakukan penilaian pada keamanan dan kepatuhan AWS Key Management Service sebagai bagian dari beberapa program kepatuhan AWS. Ini mencakup SOC, PCI, FedRAMP, HIPAA, dan sebagainya.

Topik

- Dokumen kepatuhan dan keamanan
- Pelajari selengkapnya

Dokumen kepatuhan dan keamanan

Dokumen kepatuhan dan keamanan berikut mencakup AWS KMS. Untuk melihatnya, gunakan AWS Artifact.

- Katalog Kontrol Kepatuhan Komputasi Cloud (C5)
- ISO 27001:2013 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27001:2013
- ISO 27017:2015 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27017:2015
- ISO 27018:2015 Pernyataan Penerapan (SoA)
- Sertifikasi ISO 27018:2014
- Sertifikasi ISO 9001:2015
- Pengesahan Kepatuhan (AOC) PCI DSS dan Ringkasan Tanggung Jawab
- Laporan Kontrol Organisasi Layanan (SOC) 1
- Laporan Kontrol Organisasi Layanan (SOC) 2
- Laporan Kontrol Organisasi Layanan (SOC) 2 untuk Kerahasiaan
- FedRAMP-Tinggi

Untuk bantuan menggunakan AWS Artifact, lihat Mengunduh Laporan di Artifact AWS.

Pelajari selengkapnya

Tanggung jawab kepatuhan Anda saat menggunakan AWS KMS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. Jika penggunaan AWS KMS Anda tunduk pada kepatuhan dengan standar yang diterbitkan, AWS menyediakan sumber daya untuk membantu:

- Layanan AWS Tercakup oleh Program Kepatuhan – Halaman ini membuat daftar layanan AWS yang dalam cakupan program kepatuhan spesifik. Untuk informasi umum, lihat Program Kepatuhan AWS.
- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config – Layanan AWS ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnyaAWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.

Ketahanan di AWS Key Management Service

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Availability Zone tanpa gangguan. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripadai infrastruktur pusat data tunggal atau multi tradisional.

Selain infrastruktur global AWS, AWS KMS menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan backup Anda. Untuk informasi selengkapnya tentang Wilayah Wilayah AWS dan Zona Ketersediaan, lihat Infrastruktur Global AWS.

Isolasi regional

AWS Key Management Service(AWS KMS) adalah layanan Regional mandiri yang tersedia di semua. Wilayah AWS Desain yang terisolasi secara regional AWS KMS memastikan bahwa masalah ketersediaan di satu wilayah Wilayah AWS tidak dapat mempengaruhi AWS KMS operasi di Wilayah lain mana pun. AWS KMSdirancang untuk memastikan nol waktu henti yang direncanakan, dengan semua pembaruan perangkat lunak dan operasi penskalaan dilakukan dengan mulus dan tanpa terasa.

AWS KMSService Level Agreement (SLA) mencakup komitmen layanan sebesar 99,999% untuk semua API KMS. Untuk memenuhi komitmen ini, AWS KMS memastikan bahwa semua data dan informasi otorisasi yang diperlukan untuk menjalankan permintaan API tersedia di semua host regional yang menerima permintaan tersebut.

AWS KMSInfrastruktur direplikasi di setidaknya tiga Availability Zone (AZ) di setiap Wilayah. Untuk memastikan bahwa beberapa kegagalan host tidak memengaruhi AWS KMS kinerja, AWS KMS dirancang untuk melayani lalu lintas pelanggan dari AZ mana pun di suatu Wilayah.

Perubahan yang Anda buat pada properti atau izin kunci KMS direplikasi ke semua host di Wilayah untuk memastikan bahwa permintaan berikutnya dapat diproses dengan benar oleh host mana pun di Wilayah. Permintaan untuk operasi kriptografi menggunakan kunci KMS Anda diteruskan ke armada modul keamanan AWS KMS perangkat keras (HSM), yang mana pun dapat melakukan operasi dengan kunci KMS.

Desain multi-penyewa

Desain multi-tenant AWS KMS memungkinkannya memenuhi SLA ketersediaan 99,999%, dan untuk mempertahankan tingkat permintaan yang tinggi, sekaligus melindungi kerahasiaan kunci dan data Anda.

Beberapa mekanisme penegakan integritas digunakan untuk memastikan bahwa kunci KMS yang Anda tentukan untuk operasi kriptografi selalu yang digunakan.

Materi kunci plaintext untuk kunci KMS Anda dilindungi secara luas. Materi utama dienkripsi di HSM segera setelah dibuat, dan bahan kunci terenkripsi segera dipindahkan ke penyimpanan latensi rendah yang aman. Kunci terenkripsi diambil dan didekripsi dalam HSM tepat pada waktunya untuk digunakan. Kunci plaintext tetap dalam memori HSM hanya untuk waktu yang dibutuhkan untuk menyelesaikan operasi kriptografi. Kemudian dienkripsi ulang di HSM dan kunci terenkripsi

dikembalikan ke penyimpanan. Materi kunci Plaintext tidak pernah meninggalkan HSM; itu tidak pernah ditulis ke penyimpanan persisten.

Untuk informasi selengkapnya tentang mekanisme yang AWS KMS digunakan untuk mengamankan kunci Anda, lihat Detail AWS Key Management Service Kriptografi.

Praktik terbaik ketahanan di AWS KMS

Untuk mengoptimalkan ketahanan AWS KMS sumber daya Anda, pertimbangkan strategi berikut.

- Untuk mendukung strategi pencadangan dan pemulihan bencana Anda, pertimbangkan kunci Multi-wilayah, yang merupakan kunci KMS yang dibuat dalam satu Wilayah AWS dan direplikasi hanya ke Wilayah yang Anda tentukan. Dengan kunci Multi-region, Anda dapat memindahkan sumber daya terenkripsi antara Wilayah AWS (dalam partisi yang sama) tanpa pernah mengekspos plaintext, dan mendekripsi sumber daya, bila diperlukan, di salah satu Wilayah tujuannya. Kunci Multi-wilayah terkait dapat dioperasikan karena mereka berbagi materi kunci dan ID kunci yang sama, tetapi mereka memiliki kebijakan kunci independen untuk kontrol akses resolusi tinggi. Untuk detailnya, lihat kunci Multi-Wilayah di AWS KMS.
- Untuk melindungi kunci Anda dalam layanan multi-penyewa sepertiAWS KMS, pastikan untuk menggunakan kontrol akses, termasuk kebijakan utama dan kebijakan IAM. Selain itu, Anda dapat mengirim permintaan Anda untuk AWS KMS menggunakan titik akhir antarmuka VPC yang didukung oleh. AWS PrivateLink Ketika Anda melakukannya, semua komunikasi antara VPC Amazon Anda dan AWS KMS dilakukan sepenuhnya dalam AWS jaringan menggunakan AWS KMS titik akhir khusus yang dibatasi untuk VPC Anda. Anda dapat lebih mengamankan permintaan ini dengan membuat lapisan otorisasi tambahan menggunakan kebijakan titik akhir VPC. Untuk detailnya, lihat Menghubungkan ke AWS KMS melalui titik akhir VPC.

Keamanan infrastruktur dalam AWS Key Management Service

Sebagai suatu layanan terkelola, AWS Key Management Service (AWS KMS) dilindungi oleh prosedur keamanan jaringan global AWS yang dijelaskan dalam laporan resmi Amazon Web Services: Gambaran Umum dari Proses Keamanan.

Untuk mengakses AWS KMS melalui jaringan, Anda dapat memanggil operasi AWS KMS API yang dijelaskan dalam Referensi AWS Key Management Service API. AWS KMSmembutuhkan TLS 1.2 dan merekomendasikan TLS 1.3 di semua wilayah. AWS KMSjuga mendukung TLS pasca-kuantum hibrida untuk titik akhir AWS KMS layanan di semua wilayah, kecuali Wilayah China. AWS KMStidak mendukung TLS pasca-kuantum hibrida untuk titik akhir FIPS di. AWS GovCloud

(US) Untuk menggunakan endpoint standar atau AWS KMS endpoint AWS KMS FIPS, klien harus mendukung TLS 1.2 atau yang lebih baru. Klien juga harus support suite cipher dengan Perfect Forward Secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern, misalnya Java 7 dan versi yang lebih baru, mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Anda dapat memanggil operasi API ini dari lokasi jaringan mana pun, tetapi AWS KMS mendukung kondisi kebijakan global yang memungkinkan Anda mengontrol akses ke kunci KMS berdasarkan alamat IP sumber, VPC, dan titik akhir VPC. Anda dapat menggunakan kunci kondisi ini dalam kebijakan kunci dan kebijakan IAM. Namun, kondisi ini dapat AWS mencegah penggunaan kunci KMS atas nama Anda. Untuk detailnya, lihat AWS kunci kondisi global.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan pengguna yang dapat mengambil `KMSTestRole` peran untuk menggunakan ini AWS KMS key untuk operasi kriptografi tertentu kecuali alamat IP sumber adalah salah satu alamat IP yang ditentukan dalam kebijakan.

```
{
  "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"
        ]
```

```
        }
      }
    }
}
```

Isolasi pada Host Fisik

Keamanan infrastruktur fisik yang digunakan AWS KMS tunduk pada kontrol yang dijelaskan dalam bagian Keamanan Fisik dan Lingkungan dari Amazon Web Services: Gambaran Umum Proses Keamanan. Anda dapat menemukan lebih banyak detail dalam laporan kepatuhan dan temuan audit pihak ketiga yang tercantum di bagian sebelumnya.

AWS KMS didukung oleh modul keamanan perangkat keras khusus (HSM) yang dirancang dengan kontrol khusus untuk melawan serangan fisik. HSMS adalah perangkat fisik yang tidak memiliki lapisan virtualisasi, seperti hypervisor, yang berbagi perangkat fisik di antara beberapa penyewa logis. Bahan utama untuk AWS KMS keys disimpan hanya dalam memori volatil pada HSM, dan hanya saat kunci KMS sedang digunakan. Memori ini terhapus saat HSM bergerak keluar dari keadaan operasional, termasuk shutdown dan reset yang sengaja maupun tidak disengaja. Untuk informasi mendetail tentang pengoperasian HSM AWS KMS, lihat Detail Kriptografi AWS Key Management Service.

Praktik terbaik keamanan untuk AWS Key Management Service

AWS Key Management Service(AWS KMS) mendukung banyak fitur keamanan yang dapat Anda terapkan untuk meningkatkan perlindungan kunci enkripsi Anda, termasuk kebijakan utama dan kebijakan IAM, opsi konteks enkripsi untuk operasi kriptografi pada kunci enkripsi simetris, serangkaian kunci kondisi yang ekstensif untuk menyempurnakan kebijakan utama Anda dan kebijakan IAM, dan memberikan batasan untuk membatasi hibah.

Fitur keamanan ini dijelaskan secara rinci dalam Praktik AWS Key Management Service Terbaik (PDF). Pedoman umum dalam paper teknis ini tidak mewakili solusi keamanan yang lengkap. Karena tidak semua praktik terbaik sesuai untuk semua situasi, ini tidak dimaksudkan untuk menjadi preskriptif.

Lihat juga

- Praktik terbaik untuk kebijakan IAM
- Praktik terbaik untuk AWS KMS hibah

- Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM

Kuota

Untuk membuat AWS KMS responsif dan berkinerja untuk semua pengguna, AWS KMS menerapkan dua jenis kuota, kuota sumber daya dan kuota permintaan. Setiap kuota dihitung secara independen untuk setiap Wilayah dari masing-masing Akun AWS.

Semua AWS KMS kuota dapat disesuaikan, kecuali kuota sumber daya ukuran dokumen kebijakan utama, kuota sumber daya rotasi sesuai permintaan, dan kuota permintaan penyimpanan AWS CloudHSM kunci. Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas. Untuk meminta pengurangan kuota, untuk mengubah kuota yang tidak tercantum dalam Service Quotas, atau untuk mengubah kuota di mana Wilayah AWS Service Quotas AWS KMS for tidak tersedia, silakan kunjungi AWS Support Center dan buat kasus.

Topik

Kuota sumber daya

AWS KMS menetapkan kuota sumber daya untuk memastikan bahwa itu dapat memberikan layanan yang cepat dan tangguh kepada semua pelanggan kami. Beberapa kuota sumber daya hanya berlaku untuk sumber daya yang Anda buat, tetapi tidak untuk sumber daya yang dibuat AWS layanan untuk Anda. Sumber daya yang Anda gunakan, tetapi itu tidak ada dalam Anda Akun AWS, seperti Kunci milik AWS, tidak dihitung terhadap kuota ini.

Jika Anda telah melampaui batas sumber daya, permintaan untuk membuat sumber daya tambahan dari jenis tersebut menghasilkan pesan kesalahan `LimitExceededException`.

Semua kuota AWS KMS sumber daya dapat disesuaikan, kecuali kuota ukuran dokumen kebijakan utama dan kuota sumber daya rotasi sesuai permintaan. Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas. Untuk meminta pengurangan kuota, untuk mengubah kuota yang tidak tercantum dalam Service Quotas, atau untuk mengubah kuota di mana Wilayah AWS Service Quotas AWS KMS for tidak tersedia, silakan kunjungi AWS Support Center dan buat kasus.

Tabel berikut mencantumkan dan menjelaskan kuota AWS KMS sumber daya di masing-masing Akun AWS dan Wilayah.

Nama kuota	Nilai default	Berlaku untuk	Dapat Disesuaikan
AWS KMS keys	100.000	Kunci yang dikelola pelanggan	Ya
Alias per kunci KMS	50	Alias yang dibuat pelanggan	Ya
Hibah per kunci KMS	50.000	Kunci yang dikelola pelanggan	Ya
Ukuran dokumen kebijakan utama	32 KB (32.768 byte)	Kunci yang dikelola pelanggan Kunci yang dikelola AWS	Tidak
Kuota sumber daya toko kunci kustom	10	Akun AWS dan Wilayah	Ya

Selain kuota sumber daya, AWS KMS gunakan kuota permintaan untuk memastikan daya tanggap layanan. Lihat perinciannya di the section called “Kuota permintaan”.

AWS KMS keys: 100.000

Anda dapat memiliki hingga 100.000 kunci yang dikelola pelanggan di setiap Wilayah Anda Akun AWS. Kuota ini berlaku untuk semua kunci yang dikelola pelanggan di semua Wilayah AWS terlepas dari spesifikasi kunci atau status kunci mereka. Setiap kunci KMS dianggap sebagai satu sumber daya. Kunci yang dikelola AWSdan Kunci milik AWSjangan dihitung terhadap kuota ini.

Alias per kunci KMS: 50

Anda dapat mengaitkan hingga 50 alias dengan setiap kunci yang dikelola pelanggan. Alias yang AWS berhubungan dengan Kunci yang dikelola AWStidak dihitung terhadap kuota ini. Anda mungkin mengalami kuota ini ketika membuat atau memperbarui alias.

> **Note**
>
> ResourceAliasesKondisi kms: hanya efektif jika kunci KMS sesuai dengan kuota ini. Jika kunci KMS melebihi kuota ini, kepala sekolah yang berwenang untuk menggunakan kunci KMS dengan `kms:ResourceAliases` syarat ditolak akses ke kunci KMS. Lihat perinciannya di Akses ditolak karena kuota alias.

Alias per kuota kunci KMS menggantikan kuota Alias per Wilayah yang membatasi jumlah total alias di setiap Wilayah. Akun AWS AWS KMS telah menghilangkan kuota Alias per Wilayah.

Hibah per kunci KMS: 50.000

Setiap kunci yang dikelola pelanggan dapat memiliki hingga 50.000 hibah, termasuk hibah yang dibuat oleh AWS layanan yang terintegrasi dengannya. AWS KMS Kuota ini tidak berlaku untuk Kunci yang dikelola AWSatau Kunci milik AWS.

Salah satu efek dari kuota ini adalah Anda tidak dapat melakukan lebih dari 50.000 operasi resmi hibah yang menggunakan kunci KMS yang sama pada saat yang bersamaan. Setelah Anda mencapai kuota, Anda dapat membuat hibah baru pada kunci KMS hanya ketika hibah aktif dihentikan atau dicabut.

Misalnya, ketika Anda melampirkan volume Amazon Elastic Block Store (Amazon EBS) ke instans Amazon Elastic Compute Cloud (Amazon EC2), volume didekripsi sehingga Anda dapat membacanya. Untuk mendapatkan izin untuk mendekripsi data, Amazon EBS membuat satu pemberian untuk setiap volume. Oleh karena itu, jika semua volume Amazon EBS Anda menggunakan kunci KMS yang sama, Anda tidak dapat melampirkan lebih dari 50.000 volume sekaligus.

Ukuran dokumen kebijakan kunci: 32 KB

Panjang maksimum setiap dokumen kebijakan kunci adalah 32 KB (32.768 byte). Jika Anda menggunakan dokumen kebijakan yang lebih besar untuk membuat atau memperbarui kebijakan kunci untuk kunci KMS, operasi akan gagal.

Kuota ini tidak dapat disesuaikan. Anda tidak dapat meningkatkannya dengan menggunakan Service Quotas atau dengan membuat kasus di. AWS Support Jika kebijakan kunci Anda mendekati batas, pertimbangkan untuk menggunakan pemberian, bukan pernyataan kebijakan. Pemberian sangat cocok untuk izin sementara atau sangat spesifik.

Anda menggunakan dokumen kebijakan kunci setiap kali Anda membuat atau mengubah kebijakan kunci dengan menggunakan tampilan default atau tampilan kebijakan di AWS Management Console, atau PutKeyPolicyoperasi. Kuota ini berlaku untuk dokumen kebijakan kunci Anda, meskipun Anda menggunakan tampilan default dalam konsol AWS KMS , tempat Anda tidak mengedit pernyataan JSON secara langsung.

Kuota sumber daya penyimpanan kunci kustom: 10

Anda dapat membuat hingga 10 toko kunci khusus di masing-masing Akun AWS dan Wilayah. Jika Anda mencoba membuat lebih banyak, CreateCustomKeyStoreoperasi gagal.

Kuota ini berlaku untuk jumlah total toko kunci khusus di setiap akun dan wilayah, termasuk semua toko utama dan toko AWS CloudHSM kunci eksternal, terlepas dari status koneksinya.

Rotasi sesuai permintaan: 10

Anda dapat melakukan rotasi kunci sesuai permintaan maksimal 10 kali per tombol KMS. Jika Anda mencoba melakukan lebih banyak rotasi sesuai permintaan, RotateKeyOnDemandoperasi gagal.

Kuota ini tidak dapat disesuaikan. Anda tidak dapat meningkatkannya dengan menggunakan Service Quotas atau dengan membuat kasus di. AWS Support Untuk mencegah tercapainya kuota rotasi sesuai permintaan, sebaiknya gunakan rotasi kunci otomatis bila memungkinkan.

Kuota permintaan

AWS KMS menetapkan kuota untuk jumlah operasi API yang diminta dalam setiap detik. Kuota permintaan berbeda dengan operasi API Wilayah AWS, dan faktor lainnya, seperti jenis kunci KMS. Jika Anda melebihi kuota permintaan AWS KMS API, batasi permintaan tersebut.

Semua kuota AWS KMS permintaan dapat disesuaikan, kecuali kuota permintaan toko AWS CloudHSM kunci. Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas. Untuk meminta pengurangan kuota, untuk mengubah kuota yang tidak tercantum dalam Service Quotas, atau untuk mengubah kuota di mana Wilayah AWS Service Quotas AWS KMS for tidak tersedia, silakan kunjungi AWS Support Center dan buat kasus.

Jika Anda melebihi kuota permintaan untuk GenerateDataKeyoperasi, pertimbangkan untuk menggunakan fitur caching kunci data dari. AWS Encryption SDK Menggunakan kembali kunci data dapat mengurangi frekuensi permintaan Anda. AWS KMS

Selain meminta kuota, AWS KMS menggunakan kuota sumber daya untuk memastikan kapasitas untuk semua pengguna. Lihat perinciannya di Kuota sumber daya.

Untuk melihat tren dalam tarif permintaan Anda, gunakan konsol Service Quotas. Anda juga dapat membuat CloudWatch alarm Amazon yang memberi tahu Anda ketika tingkat permintaan Anda mencapai persentase tertentu dari nilai kuota. Untuk detailnya, lihat Mengelola tarif permintaan AWS KMS API Anda menggunakan Service Quotas dan Amazon CloudWatch di Blog AWS Keamanan.

Topik

Minta kuota untuk setiap operasi AWS KMS API

Tabel ini mencantumkan kode kuota Service Quotas dan nilai default untuk setiap AWS KMS kuota permintaan. Semua kuota AWS KMS permintaan dapat disesuaikan, kecuali kuota permintaan toko AWS CloudHSM kunci.

> **Note**
>
> Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Nama kuota	Nilai default (permintaan per detik)
`Cryptographic operations (symmetric) request rate` Berlaku untuk: • `Decrypt`	Kuota bersama ini bervariasi dengan Wilayah AWS dan jenis kunci KMS yang digunakan dalam permintaan. Setiap kuota dihitung secara terpisah. • 5.500 (bersama)

Nama kuota	Nilai default (permintaan per detik)

• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
• `GenerateMac`
• `GenerateRandom`
• `ReEncrypt`
• `VerifyMac`
	• 10.000 (bersama) di Wilayah berikut:
	  • US East (Ohio) us-east-2
	  • Asia Pacific (Singapore) ap-southeast-1
	  • Asia Pacific (Sydney), ap-southeast-2
	  • Asia Pacific (Tokyo), ap-northeast-1
	  • Europe (Frankfurt), eu-central-1
	  • Europe (London), eu-west-2
	• 50.000 (bersama) di Wilayah berikut:
	  • US East (N. Virginia), us-east-1
	  • US West (Oregon), us-west-2
	  • Europe (Ireland), eu-west-1

`Cryptographic operations (RSA) request rate` Berlaku untuk: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	500 (dibagikan) untuk kunci RSA KMS

Nama kuota	Nilai default (permintaan per detik)
`Cryptographic operations (ECC and SM2) request rate` Berlaku untuk: • `Decrypt`—hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `Encrypt`—hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `ReEncrypt` —hanya didukung untuk kunci KMS SM2 (hanya Wilayah China) • `Sign` • `Verify`	300 (bersama) untuk kurva elips (ECC) dan SM2 (khusus Wilayah China) kunci KMS
`Custom key store request quotas` Berlaku untuk: • `Decrypt` • `Encrypt` • `GenerateDataKey` • `GenerateDataKeyWithoutPlaintext` • `GenerateRandom` • `ReEncrypt`	Kuota permintaan toko kunci khusus dihitung secara terpisah untuk setiap toko kunci khusus • 1.800 (dibagikan) untuk setiap toko AWS CloudHSM kunci • 1.800 (dibagikan) untuk setiap toko kunci eksternal
`CancelKeyDeletion request rate`	5
`ConnectCustomKeyStore request rate`	5
`CreateAlias request rate`	5
`CreateCustomKeyStore request rate`	5

Nama kuota	Nilai default (permintaan per detik)
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
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 Berlaku untuk: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

Nama kuota	Nilai default (permintaan per detik)
`GenerateDataKeyPair (ECC_NIST_P384) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (ECC_NIST_P521) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (ECC_SECG_P256K1) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	100
`GenerateDataKeyPair (RSA_2048) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	1

Nama kuota	Nilai default (permintaan per detik)
`GenerateDataKeyPair (RSA_3072) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,5 (1 dalam setiap interval 2 detik)
`GenerateDataKeyPair (RSA_4096) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	0,1 (1 dalam setiap interval 10 detik)
`GenerateDataKeyPair (SM2 – China Regions only) request rate` Berlaku untuk: • `GenerateDataKeyPair` • `GenerateDataKeyPairWithoutPlaintext`	25
`GetKeyPolicy request rate`	1000
`GetKeyRotationStatus request rate`	1000
`GetParametersForImport request rate`	0,25 (1 dalam setiap interval 4 detik)
`GetPublicKey request rate`	2000
`ImportKeyMaterial request rate`	5

Nama kuota	Nilai default (permintaan per detik)
`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
`ReplicateKey request rate` Operasi `ReplicateKey` dihitung sebagai satu permintaan `ReplicateKey` di Wilayah kunci primer dan dua permintaan `CreateKey` di Wilayah replika. Salah satu permintaan `CreateKey` adalah dry run untuk mendeteksi potensi masalah sebelum membuat kunci.	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

Nama kuota	Nilai default (permintaan per detik)
`UpdateCustomKeyStore request rate`	5
`UpdateKeyDescription request rate`	5
`UpdatePrimaryRegion request rate` Operasi `UpdatePrimaryRegion` dihitung sebagai dua permintaan `UpdatePrimaryRegion` ; satu permintaan dalam masing-masing dari dua Wilayah yang terkena dampak.	5

Menerapkan kuota permintaan

Saat meninjau kuota permintaan, harap ingat informasi berikut.

- Kuota permintaan berlaku untuk kunci yang dikelola pelanggan dan Kunci yang dikelola AWS. Penggunaan Kunci milik AWStidak dihitung terhadap kuota permintaan untuk Anda Akun AWS, bahkan ketika mereka digunakan untuk melindungi sumber daya di akun Anda.
- Kuota permintaan berlaku untuk permintaan yang dikirim ke titik akhir FIPS dan titik akhir bukan FIPS. Untuk daftar titik akhir AWS KMS layanan, lihat AWS Key Management Service titik akhir dan kuota di. Referensi Umum AWS
- Throttling didasarkan pada semua permintaan pada kunci KMS dari semua jenis di Wilayah. Total ini termasuk permintaan dari semua kepala sekolah di Akun AWS, termasuk permintaan dari AWS layanan atas nama Anda.
- Setiap kuota permintaan secara independen. Misalnya, permintaan untuk CreateKeyoperasi tidak berpengaruh pada kuota permintaan untuk CreateAliasoperasi. Jika permintaan `CreateAlias` Anda dibatasi, permintaan `CreateKey` masih dapat berhasil diselesaikan.
- Meskipun operasi kriptografi berbagi kuota, kuota yang dibagikan dihitung secara terpisah dari kuota untuk operasi lainnya. Misalnya, panggilan ke operasi Enkripsi dan Dekripsi berbagi kuota permintaan, tetapi kuota tersebut tidak tergantung pada kuota untuk operasi manajemen, seperti. EnableKey Misalnya, di Wilayah Eropa (London), Anda dapat melakukan 10.000 operasi kriptografi pada kunci KMS simetris ditambah 5 `EnableKey` operasi per detik tanpa dibatasi.

Kuota bersama untuk operasi kriptografis

AWS KMS operasi kriptografi berbagi kuota permintaan. Anda dapat meminta kombinasi operasi kriptografi apa pun yang didukung oleh kunci KMS, agar jumlah total operasi kriptografi tidak melebihi kuota permintaan untuk jenis kunci KMS tersebut. Pengecualian adalah GenerateDataKeyPairdan GenerateDataKeyPairWithoutPlaintext, yang berbagi kuota terpisah.

Kuota untuk berbagai jenis kunci KMS dihitung secara independen. Setiap kuota berlaku untuk semua permintaan untuk operasi ini di Akun AWS dan Wilayah dengan jenis kunci yang diberikan di setiap interval satu detik.

- Tingkat permintaan operasi kriptografi (simetris) adalah kuota permintaan bersama untuk operasi kriptografi menggunakan kunci KMS simetris di akun dan wilayah. Kuota ini berlaku untuk operasi kriptografi dengan kunci enkripsi simetris dan kunci HMAC, yang juga simetris.

  Misalnya, Anda mungkin menggunakan kunci KMS simetris Wilayah AWS dengan kuota bersama 10.000 permintaan per detik. Ketika Anda membuat 7.000 GenerateDataKeypermintaan per detik dan 2.000 permintaan Dekripsi per detik, AWS KMS tidak membatasi permintaan Anda. Namun, ketika Anda membuat 9.500 permintaan `GenerateDataKey` dan 1.000 permintaan Encrypt dan permintaan per detik, AWS KMS membatasi permintaan Anda karena melebihi kuota bersama.

  Operasi kriptografi pada kunci KMS enkripsi simetris di toko kunci kustom dihitung terhadap tingkat permintaan operasi Kriptografi (simetris) untuk akun dan kuota permintaan toko kunci khusus untuk penyimpanan kunci khusus.
- Tingkat permintaan operasi kriptografi (RSA) adalah kuota permintaan bersama untuk operasi kriptografi menggunakan kunci KMS asimetris RSA.

  Misalnya, dengan kuota permintaan 500 operasi per detik, Anda dapat membuat 200 permintaan Enkripsi dan 100 permintaan Dekripsi dengan kunci RSA KMS yang dapat mengenkripsi dan mendekripsi, ditambah 50 permintaan Tanda Tangan dan 150 permintaan Verifikasi dengan kunci RSA KMS yang dapat menandatangani dan memverifikasi.
- Tingkat permintaan operasi kriptografi (ECC) adalah kuota permintaan bersama untuk operasi kriptografi menggunakan kunci KMS asimetris kurva elips (ECC).

  Misalnya, dengan kuota permintaan 300 operasi per detik, Anda dapat membuat 100 permintaan Tanda Tangan dan 200 permintaan Verifikasi dengan kunci RSA KMS yang dapat menandatangani dan memverifikasi.

- Tingkat permintaan operasi kriptografi (khusus Wilayah SM - China) adalah kuota permintaan bersama untuk operasi kriptografi menggunakan kunci KMS asimetris SM.

  Misalnya, dengan kuota permintaan 300 operasi per detik, Anda dapat membuat 100 permintaan Enkripsi dan 100 permintaan Dekripsi dengan kunci SM2 KMS yang dapat mengenkripsi dan mendekripsi, ditambah 50 permintaan Tanda Tangan dan 50 permintaan Verifikasi dengan kunci SM2 KMS yang dapat menandatangani dan memverifikasi.
- Kuota permintaan toko kunci kustom adalah kuota permintaan bersama untuk operasi kriptografi pada kunci KMS di toko kunci kustom. Kuota ini dihitung secara terpisah untuk setiap toko kunci kustom.

  Operasi kriptografi pada kunci KMS enkripsi simetris di toko kunci kustom dihitung terhadap tingkat permintaan operasi Kriptografi (simetris) untuk akun dan kuota permintaan toko kunci khusus untuk penyimpanan kunci khusus.

Kuota untuk berbagai jenis kunci dihitung secara terpisah. Misalnya, di Wilayah Asia Pasifik (Singapura), jika Anda menggunakan kunci KMS simetris dan asimetris, Anda dapat melakukan hingga 10.000 panggilan per detik dengan kunci KMS simetris (termasuk kunci HMAC) ditambah hingga 500 panggilan tambahan per detik dengan kunci KMS asimetris RSA Anda, ditambah hingga 300 permintaan tambahan per detik dengan kunci KMS berbasis ECC Anda.

Permintaan API yang dibuat atas nama Anda

Anda dapat membuat permintaan API secara langsung atau dengan menggunakan AWS layanan terintegrasi yang membuat permintaan API atas nama Anda. AWS KMS Kuota berlaku untuk kedua jenis permintaan.

Misalnya, Anda mungkin menyimpan data di Amazon S3 menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS). Setiap kali Anda mengunggah atau mengunduh objek S3 yang dienkripsi dengan SSE-KMS, Amazon S3 `GenerateDataKey` membuat permintaan (untuk unggahan`Decrypt`) atau (untuk unduhan) atas nama Anda. AWS KMS Permintaan ini dihitung terhadap kuota Anda, jadi AWS KMS batasi permintaan jika Anda melebihi total gabungan 5.500 (atau 10.000 atau 50.000 tergantung pada Wilayah AWS) unggahan atau unduhan per detik objek S3 yang dienkripsi dengan SSE-KMS.

Permintaan lintas akun

Ketika aplikasi dalam satu Akun AWS menggunakan kunci KMS yang dimiliki oleh akun yang berbeda, itu dikenal sebagai permintaan lintas akun. Untuk permintaan lintas akun, AWS KMS batasi akun yang membuat permintaan, bukan akun yang memiliki kunci KMS. Misalnya, jika aplikasi di akun A menggunakan kunci KMS di akun B, penggunaan kunci KMS hanya berlaku untuk kuota di akun A.

Kuota permintaan toko kunci kustom

AWS KMS mempertahankan kuota permintaan untuk operasi kriptografi pada kunci KMS di toko kunci khusus. Kuota permintaan ini dihitung secara terpisah untuk setiap toko kunci kustom.

Kuota permintaan toko kunci kustom	Nilai default (permintaan per detik) untuk setiap toko kunci kustom	Dapat Disesuaikan
AWS CloudHSM kuota permintaan toko kunci	1800	Tidak
Kuota permintaan toko kunci eksternal	1800	Ya

> **Note**
>
> AWS KMS kuota permintaan toko kunci kustom tidak muncul di konsol Service Quotas. Anda tidak dapat melihat atau mengelola kuota ini dengan menggunakan operasi Service Quotas API. Untuk meminta perubahan pada kuota permintaan penyimpanan kunci eksternal Anda, kunjungi AWS Support Pusat dan buat kasus.
>
> Jika AWS CloudHSM cluster yang terkait dengan penyimpanan AWS CloudHSM kunci memproses banyak perintah, termasuk yang tidak terkait dengan penyimpanan kunci khusus, Anda mungkin mendapatkan AWS KMS `ThrottlingException` lower-than-expected tarif tertentu. Jika ini terjadi, turunkan tingkat permintaan Anda AWS KMS, kurangi beban yang tidak terkait, atau gunakan AWS CloudHSM klaster khusus untuk penyimpanan AWS CloudHSM kunci Anda.
>
> AWS KMS melaporkan pembatasan permintaan penyimpanan kunci eksternal dalam metrik. `ExternalKeyStoreThrottle` CloudWatch Anda dapat menggunakan metrik

ini untuk melihat pola pembatasan, membuat alarm, dan menyesuaikan kuota permintaan penyimpanan kunci eksternal Anda.

Permintaan untuk operasi kriptografi pada kunci KMS di toko kunci khusus diperhitungkan dalam dua kuota:

- Kuota tingkat permintaan operasi kriptografi (simetris) (per akun)

  Permintaan untuk operasi kriptografi pada kunci KMS di toko kunci kustom dihitung terhadap `Cryptographic operations (symmetric) request rate` kuota untuk masing-masing Akun AWS dan Wilayah. Misalnya, di US East (Virginia N.) (us-east-1), Akun AWS masing-masing dapat memiliki hingga 50.000 permintaan per detik pada kunci KMS enkripsi simetris, termasuk permintaan yang menggunakan kunci KMS di toko kunci kustom.

- Kuota permintaan toko kunci kustom (per toko kunci kustom)

  Permintaan untuk operasi kriptografi pada kunci KMS di toko kunci khusus juga dihitung terhadap 1.800 operasi per detik. `Custom key store request quota` Kuota ini dihitung secara terpisah untuk setiap toko kunci kustom. Mereka mungkin menyertakan permintaan dari beberapa Akun AWS yang menggunakan kunci KMS di toko kunci khusus.

Misalnya, operasi Enkripsi pada kunci KMS di toko kunci kustom (salah satu jenisnya) di Wilayah AS Timur (Virginia Utara) (us-east-1) dihitung terhadap kuota `Cryptographic operations (symmetric) request rate` tingkat akun (50.000 permintaan per detik) untuk akun dan Wilayahnya, dan menuju (1.800 permintaan per detik) untuk penyimpanan kunci kustomnya. `Custom key store request quota` Namun, permintaan untuk operasi manajemen, seperti PutKeyPolicy, pada kunci KMS di toko kunci khusus hanya berlaku untuk kuota tingkat akunnya (15 permintaan per detik).

Permintaan pelambatan AWS KMS

Untuk memastikan bahwa AWS KMS dapat memberikan respons yang cepat dan andal terhadap permintaan API dari semua pelanggan, ini membatasi permintaan API yang melebihi batas tertentu.

Pelambatan terjadi ketika AWS KMS menolak permintaan yang mungkin valid, dan mengembalikan `ThrottlingException` kesalahan seperti berikut.

```
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 throttles meminta kondisi berikut.

- Tingkat permintaan per detik melebihi kuota AWS KMS permintaan untuk akun dan Wilayah.

  Misalnya, jika pengguna di akun Anda mengirimkan 1000 `DescribeKey` permintaan dalam satu detik, AWS KMS batasi semua `DescribeKey` permintaan berikutnya dalam detik itu.

  Untuk menanggapi throttling, gunakan strategi backoff dan coba lagi. Strategi ini diterapkan secara otomatis untuk kesalahan HTTP 400 di beberapa AWS SDK.
- Permintaan tingkat tinggi yang meledak atau berkelanjutan untuk mengubah status kunci KMS yang sama. Kondisi ini sering dikenal sebagai "hot key."

  Misalnya, jika aplikasi di akun Anda mengirimkan tendangan voli persisten `EnableKey` dan `DisableKey` permintaan untuk kunci KMS yang sama, batasi permintaan AWS KMS tersebut. Pelambatan ini terjadi meskipun permintaan tidak melebihi batas request-per-second permintaan untuk operasi `EnableKey` dan`DisableKey`.

  Untuk menanggapi throttling, sesuaikan logika aplikasi Anda sehingga aplikasi hanya membuat permintaan yang diperlukan atau mengonsolidasi permintaan dari beberapa fungsi.
- Permintaan untuk operasi pada kunci KMS di penyimpanan AWS CloudHSM kunci mungkin dibatasi pada lower-than-expected tingkat ketika AWS CloudHSM cluster yang terkait dengan penyimpanan AWS CloudHSM kunci memproses banyak perintah, termasuk yang tidak terkait dengan penyimpanan kunci. AWS CloudHSM

  (AWS KMS tidak lagi membatasi permintaan untuk operasi pada kunci KMS di penyimpanan AWS CloudHSM kunci ketika tidak ada sesi PKCS #11 yang tersedia untuk klaster. AWS CloudHSM Sebaliknya, ia melempar `KMSInternalException` dan merekomendasikan agar Anda mencoba lagi permintaan Anda.)

Untuk melihat tren dalam tarif permintaan Anda, gunakan konsol Service Quotas. Anda juga dapat membuat CloudWatch alarm Amazon yang memberi tahu Anda ketika tingkat permintaan Anda mencapai persentase tertentu dari nilai kuota. Untuk detailnya, lihat Mengelola tarif permintaan AWS KMS API Anda menggunakan Service Quotas dan Amazon CloudWatch di Blog AWS Keamanan.

Semua AWS KMS kuota dapat disesuaikan, kecuali kuota sumber daya ukuran dokumen kebijakan utama, kuota sumber daya rotasi sesuai permintaan, dan kuota permintaan penyimpanan AWS CloudHSM kunci. Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas. Untuk meminta pengurangan kuota, untuk mengubah kuota yang tidak tercantum dalam Service Quotas, atau untuk mengubah kuota di mana Wilayah AWS Service Quotas AWS KMS for tidak tersedia, silakan kunjungi AWS Support Center dan buat kasus.

> **Note**
>
> AWS KMS kuota permintaan toko kunci kustom tidak muncul di konsol Service Quotas. Anda tidak dapat melihat atau mengelola kuota ini dengan menggunakan operasi Service Quotas API. Untuk meminta perubahan pada kuota permintaan penyimpanan kunci eksternal Anda, kunjungi AWS Support Pusat dan buat kasus.

Bagaimana layanan AWS menggunakan AWS KMS

Banyak layanan AWS menggunakan AWS KMS untuk mendukung enkripsi data Anda. Ketika suatu AWS layanan terintegrasiAWS KMS, Anda dapat menggunakan AWS KMS keys di akun Anda untuk melindungi data yang diterima, disimpan, atau dikelola oleh layanan untuk Anda. Untuk daftar lengkap AWS layanan yang terintegrasi dengannyaAWS KMS, lihat Integrasi AWS Layanan.

Topik berikut membahas secara rinci bagaimana layanan tertentu digunakanAWS KMS, termasuk kunci KMS yang mereka dukung, bagaimana mereka mengelola kunci data, izin yang mereka butuhkan, dan cara melacak penggunaan kunci KMS setiap layanan di akun Anda.

> **Important**
>
> AWSlayanan yang terintegrasi dengan hanya AWS KMS menggunakan kunci KMS enkripsi simetris untuk mengenkripsi data Anda. Layanan ini tidak mendukung enkripsi dengan kunci KMS asimetris. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Topik

- Bagaimana WorkSpaces menggunakan AWS KMS

Bagaimana AWS CloudTrail menggunakan AWS KMS

Anda dapat menggunakan AWS CloudTrail untuk merekam panggilan API AWS dan aktivitas lainnya untuk Akun AWS dan menyimpan informasi yang direkam untuk berkas log di bucket Amazon Simple Storage Service (Amazon S3) yang Anda pilih. Secara default, file log yang CloudTrail dimasukkan ke dalam bucket S3 Anda dienkripsi menggunakan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 (SSE-S3). Tetapi Anda dapat memilih untuk menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS). Untuk mempelajari cara mengenkripsi file CloudTrail log AndaAWS KMS, lihat Mengenkripsi File CloudTrail Log dengan AWS KMS keys (SSE-KMS) di Panduan Pengguna. AWS CloudTrail

> **Important**
>
> AWS CloudTraildan Amazon S3 hanya mendukung simetris. AWS KMS keys Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi Log Anda. CloudTrail Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Anda tidak membayar biaya penggunaan kunci saat CloudTrail membaca atau menulis file log yang dienkripsi dengan kunci SSE-KMS. Namun, Anda membayar biaya penggunaan kunci saat mengakses file CloudTrail log yang dienkripsi dengan kunci SSE-KMS. Untuk informasi selengkapnya tentang harga AWS KMS, lihat Harga AWS Key Management Service. Untuk informasi tentang CloudTrail harga, lihat AWS CloudTrailharga dan Mengelola biaya di Panduan AWS CloudTrail Pengguna.

Topik

- Memahami kapan kunci KMS Anda digunakan

Memahami kapan kunci KMS Anda digunakan

Mengenkripsi file CloudTrail log dengan AWS KMS build pada fitur Amazon S3 yang disebut enkripsi sisi server dengan (SSE-KMS). AWS KMS key Untuk mempelajari lebih lanjut tentang SSE-KMS, lihat Bagaimana Amazon Simple Storage Service (Amazon S3) menggunakan AWS KMS di panduan

ini atau Melindungi data menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Ketika Anda mengonfigurasi AWS CloudTrail untuk menggunakan SSE-KMS untuk mengenkripsi file log Anda, dan Amazon CloudTrail S3 menggunakan Anda AWS KMS keys saat Anda melakukan tindakan tertentu dengan layanan tersebut. Bagian berikut menjelaskan kapan dan bagaimana layanan tersebut dapat menggunakan kunci KMS Anda, dan memberikan informasi tambahan yang dapat Anda gunakan untuk memvalidasi penjelasan ini.

Tindakan yang menyebabkan CloudTrail dan Amazon S3 menggunakan kunci KMS Anda

- Anda mengonfigurasi CloudTrail untuk mengenkripsi file log dengan AWS KMS key
- CloudTrail menempatkan file log ke bucket S3 Anda
- Anda mendapatkan berkas log yang dienkripsi dari bucket S3 Anda

Anda mengonfigurasi CloudTrail untuk mengenkripsi file log dengan AWS KMS key

Ketika Anda memperbarui CloudTrail konfigurasi Anda untuk menggunakan kunci KMS Anda, CloudTrail mengirimkan `GenerateDataKey`permintaan AWS KMS untuk memverifikasi bahwa kunci KMS ada dan yang CloudTrail memiliki izin untuk menggunakannya untuk enkripsi. CloudTrail tidak menggunakan kunci data yang dihasilkan.

Permintaan `GenerateDataKey` tersebut mencakup informasi berikut untuk konteks enkripsi:

- Nama Sumber Daya Amazon (ARN) dari jalan setapak CloudTrail
- ARN dari bucket S3 dan jalur tempat file CloudTrail log dikirim

`GenerateDataKey`Permintaan menghasilkan entri di CloudTrail log Anda yang mirip dengan contoh berikut. Ketika Anda melihat entri log seperti ini, Anda dapat menentukan bahwa CloudTrail

(1)
disebut AWS KMS

(2)
`GenerateDataKey` operation

(3)
untuk trail

(4)

tertentu. AWS KMSmembuat kunci data di bawah kunci KMS tertentu
(5).

> **Note**
>
> Anda mungkin perlu menggulir ke kanan untuk melihat beberapa panggilan dalam entri log contoh berikut.

```
{
  "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 menempatkan file log ke bucket S3 Anda

Setiap kali CloudTrail memasukkan file log ke bucket S3 Anda, Amazon S3 mengirimkan `GenerateDataKey`permintaan AWS KMS ke atas nama. CloudTrail Menanggapi permintaan ini, AWS KMS buat kunci data unik dan kemudian mengirimkan Amazon S3 dua salinan kunci data, satu dalam teks biasa dan satu yang dienkripsi dengan kunci KMS yang ditentukan. Amazon S3 menggunakan kunci data plaintext untuk mengenkripsi file CloudTrail log dan kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan. Amazon S3 menyimpan kunci data terenkripsi sebagai metadata dengan file log terenkripsi. CloudTrail

Permintaan `GenerateDataKey` tersebut mencakup informasi berikut untuk konteks enkripsi:

- Nama Sumber Daya Amazon (ARN) dari jalan setapak CloudTrail
- ARN dari objek S3 (file log) CloudTrail

Setiap `GenerateDataKey` permintaan menghasilkan entri di CloudTrail log Anda yang mirip dengan contoh berikut. Ketika Anda melihat entri log seperti ini, Anda dapat menentukan bahwa CloudTrail

(1)

disebut AWS KMS

(2)

`GenerateDataKey` operation

(3)

untuk trail

(4))
tertentu untuk melindungi file log tertentu
(5)).
AWS KMSmembuat kunci data di bawah kunci KMS yang ditentukan
(6)),
ditampilkan dua kali dalam entri log yang sama.

> **Note**
>
> Anda mungkin perlu menggulir ke kanan untuk melihat beberapa panggilan dalam entri log contoh berikut.

```
{
  "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"
}
```

Anda mendapatkan berkas log yang dienkripsi dari bucket S3 Anda

Setiap kali Anda mendapatkan file CloudTrail log terenkripsi dari bucket S3 Anda, Amazon S3 mengirimkan `Decrypt`permintaan ke AWS KMS atas nama Anda untuk mendekripsi kunci data terenkripsi file log. Menanggapi permintaan ini, AWS KMS gunakan kunci KMS Anda untuk mendekripsi kunci data dan kemudian mengirimkan kunci data teks biasa ke Amazon S3. Amazon S3 menggunakan kunci data plaintext untuk mendekripsi file CloudTrail log dan kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.

Permintaan `Decrypt` tersebut mencakup informasi berikut untuk konteks enkripsi:

- Nama Sumber Daya Amazon (ARN) dari jalan setapak CloudTrail
- ARN dari objek S3 (file log) CloudTrail

Setiap `Decrypt` permintaan menghasilkan entri di CloudTrail log Anda yang mirip dengan contoh berikut. Ketika Anda melihat entri log seperti ini, Anda dapat menentukan bahwa pengguna di Akun AWS (1) memanggil AWS KMS (2) `Decrypt` operation (3) untuk trail (4) tertentu dan file log tertentu (5). AWS KMSmendekripsi kunci data di bawah kunci KMS tertentu (). 6

> **Note**
>
> Anda mungkin perlu menggulir ke kanan untuk melihat beberapa panggilan dalam entri log contoh berikut.

```
{
  "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"
}
```

Bagaimana Amazon DynamoDB menggunakan AWS KMS

Amazon DynamoDB adalah layanan database NoSQL yang dikelola sepenuhnya. DynamoDB terintegrasi dengan AWS Key Management Service (AWS KMS) untuk mendukung fitur enkripsi sisi server enkripsi saat diam.

Dengan enkripsi saat diam, DynamoDB secara transparan mengenkripsi semua data pelanggan dalam tabel DynamoDB, termasuk kunci utama dan indeks sekunder lokal dan global, setiap kali

tabel dilanjutkan ke disk. (Jika tabel Anda memiliki kunci sortir, beberapa kunci sortir yang menandai batas kisaran disimpan dalam plaintext dalam metadata tabel.) Ketika Anda mengakses tabel Anda, DynamoDB mendekripsi data tabel secara transparan. Anda tidak perlu mengubah aplikasi Anda untuk menggunakan atau mengelola tabel terenkripsi.

Enkripsi saat istirahat juga melindungi DynamoDB streams, tabel global, dan cadangan setiap kali objek tersebut disimpan ke media tahan lama. Pernyataan tentang tabel dalam topik ini juga berlaku untuk objek ini.

Semua tabel DynamoDB dienkripsi. Tidak ada pilihan untuk mengaktifkan atau menonaktifkan enkripsi untuk tabel baru atau yang sudah ada. Secara default, semua tabel dienkripsi di bawah akun Kunci milik AWS layanan DynamoDB. Namun, Anda dapat memilih opsi untuk mengenkripsi beberapa atau semua tabel Anda di bawah kunci yang dikelola pelanggan atau Kunci yang dikelola AWSuntuk DynamoDB di akun Anda.

Untuk detail tentang dukungan Amazon DynamoDB untuk kunci KMS, lihat enkripsi DynamoDB saat istirahat di Panduan Pengembang Amazon DynamoDB.

Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS

Topik ini membahas secara mendetail bagaimana Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS untuk mengenkripsi volume dan snapshot. Untuk petunjuk dasar tentang mengenkripsi volume Amazon EBS, lihat Enkripsi Amazon EBS.

Topik

- Enkripsi Amazon EBS
- Menggunakan kunci KMS dan kunci data
- Konteks enkripsi Amazon EBS
- Mendeteksi kegagalan Amazon EBS
- Menggunakan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi

Enkripsi Amazon EBS

Saat Anda melampirkan volume Amazon EBS terenkripsi ke jenis instans yang didukung Amazon Elastic Compute Cloud (Amazon EC2), data disimpan saat istirahat pada volume, disk I/O, dan

snapshot yang dibuat dari volume semuanya dienkripsi. Enkripsi terjadi pada server yang menjadi host instans Amazon EC2.

Fitur ini didukung di semua jenis volume Amazon EBS. Anda mengakses volume terenkripsi dengan cara yang sama Anda mengakses volume lain; enkripsi dan dekripsi ditangani secara transparan dan mereka tidak memerlukan tindakan tambahan dari Anda, instans EC2 Anda, atau aplikasi Anda. Snapshot volume terenkripsi dienkripsi secara otomatis, dan volume yang dibuat dari snapshot terenkripsi juga dienkripsi secara otomatis.

Status enkripsi volume EBS ditentukan saat Anda membuat volume. Anda tidak dapat mengubah status enkripsi dari volume yang sudah ada. Namun, Anda dapat memigrasi data antara volume terenkripsi dan tidak terenkripsi dan menerapkan status enkripsi baru saat menyalin snapshot.

Amazon EBS mendukung enkripsi opsional secara default. Anda dapat mengaktifkan enkripsi secara otomatis pada semua volume EBS baru dan salinan snapshot di Akun AWS dan Wilayah Anda. Pengaturan konfigurasi ini tidak memengaruhi volume atau snapshot yang sudah ada. Untuk detailnya, lihat Enkripsi secara default di Panduan Pengguna Amazon EC2 untuk Instans Linux atau Panduan Pengguna Amazon EC2 untuk Windows Instances.

Menggunakan kunci KMS dan kunci data

Saat Anda membuat volume Amazon EBS terenkripsi, Anda menentukan file. AWS KMS key Secara default, Amazon EBS menggunakan Kunci yang dikelola AWSuntuk Amazon EBS di akun Anda ()`aws/ebs`. Namun, Anda dapat menentukan kunci yang dikelola pelanggan yang Anda buat dan kelola.

Untuk menggunakan kunci yang dikelola pelanggan, Anda harus memberikan izin Amazon EBS untuk menggunakan kunci KMS atas nama Anda. Untuk daftar izin yang diperlukan, lihat Izin untuk pengguna IAM di Panduan Pengguna Amazon EC2 untuk Instans Linux atau Panduan Pengguna Amazon EC2 untuk Windows Instances.

> **Important**
>
> Amazon EBS hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi volume Amazon EBS. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Untuk setiap volume, Amazon EBS meminta AWS KMS untuk menghasilkan kunci data unik yang dienkripsi di bawah kunci KMS yang Anda tentukan. Amazon EBS menyimpan kunci data terenkripsi

dengan volume. Kemudian, ketika Anda melampirkan volume instans Amazon EC2, Amazon EBS memanggil AWS KMS untuk mendekripsi kunci data. Amazon EBS menggunakan kunci data plaintext dalam memori hipervisor untuk mengenkripsi semua disk I/O ke volume. Untuk detailnya, lihat Bagaimana enkripsi EBS bekerja di Panduan Pengguna Amazon EC2 untuk Instans Linux atau Panduan Pengguna Amazon EC2 untuk Instans Windows.

Konteks enkripsi Amazon EBS

Dalam permintaannya GenerateDataKeyWithoutPlaintextdan Dekripsi ke, AWS KMS Amazon EBS menggunakan konteks enkripsi dengan pasangan nama-nilai yang mengidentifikasi volume atau snapshot dalam permintaan. Nama dalam konteks enkripsi tidak berbeda.

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS secara kriptografi mengikat konteks enkripsi untuk data terenkripsi tersebut. Untuk mendekripsi data, Anda harus meneruskan konteks enkripsi yang sama.

Untuk semua volume dan untuk snapshot terenkripsi yang dibuat dengan operasi Amazon EBS CreateSnapshot, Amazon EBS menggunakan ID volume sebagai nilai konteks enkripsi. Di `requestParameters` bidang entri CloudTrail log, konteks enkripsi terlihat mirip dengan yang berikut:

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Untuk snapshot terenkripsi yang dibuat dengan CopySnapshotoperasi Amazon EC2, Amazon EBS menggunakan ID snapshot sebagai nilai konteks enkripsi. Di `requestParameters` bidang entri CloudTrail log, konteks enkripsi terlihat mirip dengan yang berikut:

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Mendeteksi kegagalan Amazon EBS

Untuk membuat volume EBS terenkripsi atau melampirkan volume ke instans EC2, Amazon EBS dan infrastruktur Amazon EC2 harus dapat menggunakan kunci KMS yang Anda tentukan untuk

enkripsi volume EBS. Ketika kunci KMS tidak dapat digunakan—misalnya, ketika status kuncinya tidak `Enabled` —pembuatan volume atau lampiran volume gagal.

Dalam hal ini, Amazon EBS mengirimkan acara ke Amazon EventBridge (sebelumnya CloudWatch Acara) untuk memberi tahu Anda tentang kegagalan tersebut. Di EventBridge, Anda dapat menetapkan aturan yang memicu tindakan otomatis dalam menanggapi peristiwa ini. Untuk informasi selengkapnya, lihat CloudWatch Acara Amazon untuk Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux, terutama bagian berikut:

- Kunci Enkripsi Tidak Valid pada Volume Lampirkan atau Pasang Kembali
- Kunci Enkripsi Tidak Valid pada Buat Volume

Untuk memperbaiki kegagalan ini, pastikan kunci KMS yang Anda tentukan untuk enkripsi volume EBS diaktifkan. Untuk melakukan ini, pertama-tama lihat kunci KMS untuk menentukan status kunci saat ini (kolom Status diAWS Management Console). Kemudian, lihat informasi di salah satu tautan berikut:

- Jika status kunci kunci KMS dinonaktifkan, aktifkan.
- Jika status kunci kunci KMS sedang menunggu impor, impor materi kunci.
- Jika status kunci kunci KMS tertunda penghapusan, batalkan penghapusan kunci.

Menggunakan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi

Anda dapat menggunakan AWS CloudFormation untuk membuat volume Amazon EBS terenkripsi. Untuk informasi lebih lanjut, lihat AWS::EC2::Volume dalam Panduan PenggunaAWS CloudFormation.

Bagaimana Amazon Elastic Transcoder menggunakan AWS KMS

Anda dapat menggunakan Amazon Elastic Transcoder untuk mengonversi file media yang Anda simpan di bucket Amazon S3 ke format yang dibutuhkan oleh perangkat pemutaran konsumen. Kedua file input dan output dapat dienkripsi dan didekripsi. Bagian berikut membahas bagaimana cara AWS KMS digunakan untuk kedua proses.

Topik

- Mengenkripsi file input
- Mendekripsi file input
- Mengenkripsi file output
- Perlindungan konten HLS
- Konteks enkripsi Elastic Transcoder

Mengenkripsi file input

Sebelum Anda dapat menggunakan Elastic Transcoder, Anda harus membuat bucket Amazon S3 dan mengunggah file media Anda ke dalamnya. Anda dapat mengenkripsi file sebelum mengunggah dengan menggunakan enkripsi sisi klien AES atau setelah mengunggah dengan menggunakan enkripsi sisi server Amazon S3.

Jika Anda memilih enkripsi sisi klien menggunakan AES, Anda bertanggung jawab untuk mengenkripsi file sebelum mengunggahnya ke Amazon S3, dan Anda harus memberikan akses Elastic Transcoder ke kunci enkripsi. Anda melakukan ini dengan menggunakan simetris AWS KMSAWS KMS keyuntuk melindungi kunci enkripsi AES yang Anda gunakan untuk mengenkripsi file media.

Jika Anda memilih enkripsi sisi server, Anda mengizinkan Amazon S3 untuk mengenkripsi dan mendekripsi semua file atas nama Anda. Anda dapat mengonfigurasi Amazon S3 untuk menggunakan salah satu dari tiga jenis kunci enkripsi yang berbeda untuk melindungi kunci data unik yang mengenkripsi file Anda:

- Kunci Amazon S3, kunci enkripsi yang dimiliki dan dikelola Amazon S3. Ini bukan bagian dari Akun AWS Anda.
- Kunci yang dikelola AWSUntuk Amazon S3, kunci KMS yang merupakan bagian dari akun Anda, tetapi dibuat dan dikelola oleh AWS
- Setiap kunci terkelola pelanggan simetris yang Anda buat dengan menggunakan AWS KMS

> **Important**
>
> Untuk enkripsi sisi klien dan sisi server, Elastic Transcoder hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi file Elastic

Transcoder Anda. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Anda dapat mengaktifkan enkripsi dan menentukan kunci dengan menggunakan konsol Amazon S3 atau API Amazon S3 yang sesuai. Untuk informasi selengkapnya tentang cara Amazon S3 melakukan enkripsi, lihat Melindungi data menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Saat Anda melindungi file input dengan menggunakan Kunci yang dikelola AWS untuk Amazon S3 di akun atau kunci yang dikelola pelanggan, Amazon S3 AWS KMS dan berinteraksi dengan cara berikut:

1. Amazon S3 meminta kunci data plaintext dan salinan kunci data yang dienkripsi di bawah kunci KMS yang ditentukan.
2. AWS KMSmembuat kunci data, mengenkripsinya dengan kunci KMS yang ditentukan, dan kemudian mengirimkan kunci data teks biasa dan kunci data terenkripsi ke Amazon S3.
3. Amazon S3 menggunakan kunci data plaintext untuk mengenkripsi file media kemudian menyimpan file dalam bucket Amazon S3 yang ditentukan.
4. Amazon S3 menyimpan kunci data terenkripsi bersama file media terenkripsi.

Mendekripsi file input

Jika Anda memilih enkripsi sisi server Amazon S3 untuk mengenkripsi file input, Elastic Transcoder tidak mendekripsi file. Sebaliknya, Elastic Transcoder bergantung pada Amazon S3 untuk melakukan dekripsi tergantung pada pengaturan yang Anda tentukan saat membuat tugas dan alur.

Kombinasi pengaturan berikut tersedia.

Mode enkripsi	kunci AWS KMS	Arti
S3	Default	Amazon S3 membuat dan mengelola kunci yang digunakan untuk mengenkripsi dan mendekripsi file media. Prosesnya buram bagi pengguna.

Mode enkripsi	kunci AWS KMS	Arti
`S3-AWS-KMS`	Default	Amazon S3 menggunakan kunci data yang dienkripsi secara default untuk Amazon Kunci yang dikelola AWS S3 di akun Anda untuk mengenkripsi file media.
`S3-AWS-KMS`	Kustom (dengan ARN)	Amazon S3 menggunakan kunci data yang dienkripsi oleh kunci terkelola pelanggan yang ditentukan untuk mengenkripsi file media.

Saat `S3-AWS-KMS` ditentukan, Amazon S3 dan AWS KMS bekerja sama dengan cara berikut untuk melakukan dekripsi.

1. Amazon S3 mengirimkan kunci data terenkripsi ke AWS KMS.
2. AWS KMSmendekripsi kunci data dengan menggunakan kunci KMS yang sesuai, dan kemudian mengirimkan kunci data teks biasa kembali ke Amazon S3.
3. Amazon S3 menggunakan kunci data plaintext untuk mendekripsi ciphertext.

Jika Anda memilih enkripsi sisi klien menggunakan kunci AES, Elastic Transcoder mengambil file terenkripsi dari bucket Amazon S3 dan mendekripsinya. Elastic Transcoder menggunakan kunci KMS yang Anda tentukan saat membuat pipeline untuk mendekripsi kunci AES dan kemudian menggunakan kunci AES untuk mendekripsi file media.

Mengenkripsi file output

Elastic Transcoder mengenkripsi file output tergantung pada bagaimana Anda menentukan pengaturan enkripsi ketika Anda membuat tugas dan pipa. Pilihan berikut tersedia.

Mode enkripsi	kunci AWS KMS	Arti
S3	Default	Amazon S3 membuat dan mengelola kunci yang digunakan untuk mengenkripsi file output.
S3-AWS-KMS	Default	Amazon S3 menggunakan kunci data yang dibuat oleh AWS KMS dan dienkripsi oleh untuk Amazon Kunci yang dikelola AWS S3 di akun Anda.
S3-AWS-KMS	Kustom (dengan ARN)	Amazon S3 menggunakan kunci data yang dienkripsi dengan menggunakan kunci terkelola pelanggan yang ditentukan oleh ARN untuk mengenkripsi file media.
AES-	Default	Elastic Transcoder menggunakan untuk Kunci yang dikelola AWS Amazon S3 di akun Anda untuk mendekripsi kunci AES tertentu yang Anda berikan dan menggunakan kunci tersebut untuk mengenkripsi file output.
AES-	Kustom (dengan ARN)	Elastic Transcoder menggunakan kunci terkelola pelanggan yang ditentukan oleh ARN untuk mendekrip si kunci AES tertentu yang Anda berikan dan menggunak

Mode enkripsi	kunci AWS KMS	Arti
		an kunci tersebut untuk mengenkripsi file output.

Bila Anda menentukan bahwa Kunci yang dikelola AWS untuk Amazon S3 di akun Anda atau kunci yang dikelola pelanggan digunakan untuk mengenkripsi file output, Amazon S3 dan AWS KMS berinteraksi dengan cara berikut:

1. Amazon S3 meminta kunci data plaintext dan salinan kunci data yang dienkripsi di bawah kunci KMS yang ditentukan.
2. AWS KMSmembuat kunci data, mengenkripsinya di bawah kunci KMS, dan mengirimkan kunci data teks biasa dan kunci data terenkripsi ke Amazon S3.
3. Amazon S3 mengenkripsi media menggunkaan kunci data dan menyimpannya dalam bucket Amazon S3 yang ditentukan.
4. Amazon S3 menyimpan kunci data terenkripsi bersama file media terenkripsi.

Saat Anda menentukan bahwa kunci AES yang Anda berikan digunakan untuk mengenkripsi file output, kunci AES harus dienkripsi menggunakan kunci KMS. AWS KMS Elastic Transcoder, AWS KMS, dan Anda berinteraksi dengan cara berikut:

1. Anda mengenkripsi kunci AES Anda dengan memanggil operasi Encrypt di API. AWS KMS AWS KMSmengenkripsi kunci dengan menggunakan kunci KMS yang ditentukan. Anda menentukan kunci KMS mana yang akan digunakan saat Anda membuat pipeline.
2. Anda menentukan file yang berisi kunci AES dienkripsi ketika Anda membuat tugas Elastic Transcoder.
3. Elastic Transcoder mendekripsi kunci dengan memanggil operasi Dekripsi di API AWS KMS, melewati kunci terenkripsi sebagai ciphertext.
4. Elastic Transcoder menggunakan kunci AES didekripsi untuk mengenkripsi file media output kemudian menghapus kunci AES didekripsi dari memori. Hanya salinan terenkripsi yang awalnya Anda tetapkan dalam tugas yang disimpan ke disk.
5. Anda dapat mengunduh file output terenkripsi dan mendekripsinya secara lokal dengan menggunakan kunci AES asli yang Anda tetapkan.

> **Important**
>
> AWS tidak pernah menyimpan kunci enkripsi privat Anda. Oleh karena itu, penting bagi Anda untuk mengelola kunci Anda dengan aman. Jika Anda menghilangkannya, Anda tidak dapat mendekripsi data Anda.

Perlindungan konten HLS

HTTP Live Streaming (HLS) adalah protokol streaming adaptif. Elastic Transcoder mendukung HLS dengan memecah file input Anda menjadi file individual yang lebih kecil yang disebut segmen media. Satu set segmen media individual yang sesuai berisi materi yang sama yang dikodekan pada laju bit yang berbeda, sehingga memungkinkan pemain untuk memilih stream yang paling sesuai dengan bandwidth yang tersedia. Elastic Transcoder juga membuat daftar putar yang berisi metadata untuk berbagai segmen yang tersedia untuk streaming.

Saat Anda mengaktifkan perlindungan konten HLS, setiap segmen media dienkripsi menggunakan kunci enkripsi AES 128-bit. Saat konten dilihat, selama proses pemutaran, pemutar mengunduh kunci dan mendekripsi segmen media.

Dua jenis kunci yang digunakan: kunci KMS dan kunci data. Anda harus membuat kunci KMS untuk digunakan untuk mengenkripsi dan mendekripsi kunci data. Elastic Transcoder menggunakan kunci data untuk mengenkripsi dan mendekripsi segmen media. Kunci data harus AES-128. Semua variasi dan segmen konten yang sama dienkripsi menggunakan kunci data yang sama. Anda dapat memberikan kunci data atau meminta Elastic Transcoder membuatnya untuk Anda.

Kunci KMS dapat digunakan untuk mengenkripsi kunci data pada titik-titik berikut:

- Jika Anda menyediakan kunci data Anda sendiri, Anda harus mengenkripsi sebelum meneruskannya ke Elastic Transcoder.
- Jika Anda meminta Elastic Transcoder menghasilkan kunci data, maka Elastic Transcoder mengenkripsi kunci data untuk Anda.

Kunci KMS dapat digunakan untuk mendekripsi kunci data pada titik-titik berikut:

- Elastic Transcoder mendekripsi kunci data yang Anda disediakan ketika perlu menggunakan kunci data untuk mengenkripsi file output atau mendekripsi file input.

- Anda mendekripsi kunci data yang dihasilkan oleh Elastic Transcoder dan menggunakannya untuk mendekripsi file output.

Untuk informasi selengkapnya, lihat Perlindungan Konten HLS di Panduan Developer Amazon Elastic Transcoder.

Konteks enkripsi Elastic Transcoder

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS secara kriptografi mengikat konteks enkripsi untuk data terenkripsi tersebut. Untuk mendekripsi data, Anda harus lulus dalam konteks enkripsi yang sama.

Elastic Transcoder menggunakan konteks enkripsi yang sama di semua permintaan API AWS KMS untuk menghasilkan kunci data, mengenkripsi, dan mendekripsi.

```
"service" : "elastictranscoder.amazonaws.com"
```

Konteks enkripsi ditulis ke CloudTrail log untuk membantu Anda memahami bagaimana kunci AWS KMS KMS yang diberikan digunakan. Di `requestParameters` bidang file CloudTrail log, konteks enkripsi terlihat mirip dengan yang berikut ini:

```
"encryptionContext": {
  "service" : "elastictranscoder.amazonaws.com"
}
```

Untuk informasi lebih lanjut tentang cara mengonfigurasi tugas Elastic Transcoder untuk menggunakan salah satu opsi enkripsi yang didukung, lihat Opsi Enkripsi Data di Panduan Developer Amazon Elastic Transcoder.

Bagaimana Amazon EMR menggunakan AWS KMS

Saat Anda menggunakan klaster Amazon EMR, Anda dapat mengonfigurasi klaster untuk mengenkripsi data at rest sebelum menyimpannya ke lokasi penyimpanan tetap. Anda dapat mengenkripsi data at rest pada Sistem File EMR (EMRFS), volume penyimpanan simpul klaster, atau keduanya. Untuk mengenkripsi data saat istirahat, Anda dapat menggunakan fileAWS KMS key. Topik berikut menjelaskan bagaimana cluster EMR Amazon menggunakan kunci KMS untuk mengenkripsi data saat istirahat.

> **Important**
>
> Amazon EMR hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi data saat istirahat di cluster EMR Amazon. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Klaster Amazon EMR juga mengenkripsi data dalam transit, yang berarti klaster mengenkripsi data sebelum mengirimnya melalui jaringan. Anda tidak dapat menggunakan kunci KMS untuk mengenkripsi data dalam perjalanan. Untuk informasi selengkapnya, lihat Enkripsi Data Dalam Transit di Panduan Pengelolaan Amazon EMR.

Untuk informasi selengkapnya tentang semua opsi enkripsi yang tersedia di Amazon EMR, lihat Opsi Enkripsi dalam Panduan Manajemen Amazon EMR.

Topik

- Mengenkripsi data pada sistem file EMR (EMRFS)
- Mengenkripsi data pada volume penyimpanan simpul klaster
- Konteks enkripsi

Mengenkripsi data pada sistem file EMR (EMRFS)

Klaster Amazon EMR menggunakan dua sistem file terdistribusi:

- Sistem File Terdistribusi Hadoop (HDFS). Enkripsi HDFS tidak menggunakan kunci KMS. AWS KMS
- Sistem File EMR (EMRFS). Amazon EMR adalah implementasi HDFS yang memungkinkan klaster Amazon EMR untuk menyimpan data dalam Amazon Simple Storage Service (Amazon S3). EMRFS mendukung empat opsi enkripsi, dua di antaranya menggunakan kunci KMS. AWS KMS Untuk informasi selengkapnya tentang keempat opsi enkripsi EMRFS, lihat Opsi Enkripsi dalam Panduan Manajemen Amazon EMR.

Dua opsi enkripsi EMRFS yang menggunakan kunci KMS menggunakan fitur enkripsi berikut yang ditawarkan oleh Amazon S3:

- Melindungi data menggunakan enkripsi sisi server dengan AWS Key Management Service (SSE-KMS). Cluster EMR Amazon mengirimkan data ke Amazon S3. Amazon S3 menggunakan kunci KMS untuk mengenkripsi data sebelum menyimpannya ke bucket S3. Untuk informasi selengkapnya tentang cara kerjanya, lihat Proses untuk mengenkripsi data pada EMRFS dengan SSE-KMS.
- Melindungi data menggunakan enkripsi sisi klien (CSE-KMS). Data dalam EMR Amazon dienkripsi di bawah AWS KMS key sebelum dikirim ke Amazon S3 untuk penyimpanan. Untuk informasi selengkapnya tentang cara kerjanya, lihat Proses untuk mengenkripsi data pada EMRFS dengan CSE-KMS.

Saat Anda mengonfigurasi kluster EMR Amazon untuk mengenkripsi data pada EMRFS dengan kunci KMS, Anda memilih kunci KMS yang ingin digunakan Amazon S3 atau kluster EMR Amazon. Dengan SSE-KMS, Anda dapat memilih untuk Amazon Kunci yang dikelola AWS S3 dengan alias aws/s3, atau kunci terkelola pelanggan simetris yang Anda buat. Dengan enkripsi sisi klien, Anda harus memilih kunci terkelola pelanggan simetris yang Anda buat. Ketika Anda memilih kunci yang dikelola pelanggan, Anda harus memastikan bahwa klaster EMR Amazon Anda memiliki izin untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Menggunakan AWS KMS keys enkripsi di Panduan Manajemen EMR Amazon.

Untuk enkripsi sisi server dan sisi klien, kunci KMS yang Anda pilih adalah kunci root dalam alur kerja enkripsi amplop. Data dienkripsi dengan kunci data unik yang dienkripsi di bawah kunci KMS. AWS KMS Data terenkripsi dan salinan terenkripsi dari kunci datanya disimpan bersama sebagai objek terenkripsi tunggal dalam bucket S3. Untuk informasi lebih lanjut tentang cara kerjanya, lihat topik berikut.

Topik

- Proses untuk mengenkripsi data pada EMRFS dengan SSE-KMS
- Proses untuk mengenkripsi data pada EMRFS dengan CSE-KMS

Proses untuk mengenkripsi data pada EMRFS dengan SSE-KMS

Saat Anda mengonfigurasi klaster Amazon EMR untuk menggunakan SSE-KMS, proses enkripsi bekerja seperti ini:

1. Klaster mengirimkan data ke Amazon S3 untuk penyimpanan dalam bucket S3.

2. Amazon S3 mengirimkan GenerateDataKeypermintaan keAWS KMS, menentukan ID kunci kunci KMS yang Anda pilih saat Anda mengonfigurasi cluster untuk menggunakan SSE-KMS. Permintaan mencakup konteks enkripsi; untuk informasi selengkapnya, lihat Konteks enkripsi.
3. AWS KMS menghasilkan kunci enkripsi data yang unik (kunci data) kemudian mengirimkan dua salinan kunci data ini ke Amazon S3. Satu salinan tidak terenkripsi (plaintext), dan salinan lainnya dienkripsi di bawah kunci KMS.
4. Amazon S3 menggunakan kunci data plaintext untuk mengenkripsi data yang diterima di langkah 1, kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.
5. Amazon S3 menyimpan data terenkripsi dan salinan terenkripsi dari kunci data bersama sebagai objek terenkripsi tunggal dalam bucket S3.

Proses dekripsi bekerja seperti ini:

1. Klaster mengajukan permintaan objek data yang dienkripsi dari bucket S3.
2. Amazon S3 mengekstrak kunci data terenkripsi dari objek S3, dan kemudian mengirimkan kunci data terenkripsi ke AWS KMS dengan permintaan Dekripsi. Permintaan mencakup konteks enkripsi.
3. AWS KMSmendekripsi kunci data terenkripsi menggunakan kunci KMS yang sama yang digunakan untuk mengenkripsinya, dan kemudian mengirimkan kunci data yang didekripsi (teks biasa) ke Amazon S3.
4. Amazon S3 menggunakan kunci data plaintext untuk mendekripsi data yang dienkripsi, kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.
5. Amazon S3 mengirimkan data didekripsi ke klaster.

Proses untuk mengenkripsi data pada EMRFS dengan CSE-KMS

Saat Anda mengonfigurasi klaster Amazon EMR untuk menggunakan CSE-KMS, proses enkripsi bekerja seperti ini:

1. Saat siap untuk menyimpan data di Amazon S3, klaster mengirimkan GenerateDataKeypermintaan keAWS KMS, menentukan ID kunci kunci KMS yang Anda pilih saat Anda mengonfigurasi cluster untuk menggunakan CSE-KMS. Permintaan mencakup konteks enkripsi; untuk informasi selengkapnya, lihat Konteks enkripsi.

2. AWS KMS menghasilkan kunci enkripsi data yang unik (kunci data) kemudian mengirimkan dua salinan kunci data ini ke klaster. Satu salinan tidak terenkripsi (plaintext), dan salinan lainnya dienkripsi di bawah kunci KMS.
3. Klaster menggunakan kunci data plaintext untuk mengenkripsi data, kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.
4. Klaster mengombinasikan data terenkripsi dan salinan terenkripsi dari kunci data bersama sebagai objek terenkripsi tunggal.
5. Klaster mengirimkan objek dienkripsi ke Amazon S3 untuk penyimpanan.

Proses dekripsi bekerja seperti ini:

1. Klaster mengajukan permintaan objek data yang dienkripsi dari bucket S3.
2. Amazon S3 mengirimkan objek terenkripsi ke klaster.
3. Klaster mengekstrak kunci data terenkripsi dari objek terenkripsi, dan kemudian mengirimkan kunci data terenkripsi ke AWS KMS dengan permintaan Dekripsi. Permintaan mencakup konteks enkripsi.
4. AWS KMSmendekripsi kunci data terenkripsi menggunakan kunci KMS yang sama yang digunakan untuk mengenkripsinya, dan kemudian mengirimkan kunci data yang didekripsi (plaintext) ke cluster.
5. Klaster menggunakan kunci data plaintext untuk mendekripsi data terenkripsi, kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.

Mengenkripsi data pada volume penyimpanan simpul klaster

Klaster Amazon EMR adalah koleksi instans Amazon Elastic Compute Cloud (Amazon EC2). Setiap instans dalam klaster disebut simpul klaster atau simpul. Setiap simpul dapat memiliki dua jenis volume penyimpanan: volume penyimpanan instans, dan volume Amazon Elastic Block Store (Amazon EBS). Anda dapat mengonfigurasi klaster untuk menggunakan Penyiapan Kunci Terpadu Linux (LUKS) untuk mengenkripsi kedua jenis volume penyimpanan pada simpul (tetapi bukan boot volume dari setiap simpul). Ini disebut enkripsi disk lokal.

Saat Anda mengaktifkan enkripsi disk lokal untuk sebuah cluster, Anda dapat memilih untuk mengenkripsi kunci LUKS dengan kunci KMS. AWS KMS Anda harus memilih kunci yang dikelola pelanggan yang Anda buat; Anda tidak dapat menggunakan Kunci yang dikelola AWS. Jika Anda memilih kunci yang dikelola pelanggan, Anda harus memastikan bahwa klaster EMR Amazon Anda

memiliki izin untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Menggunakan AWS KMS keys enkripsi di Panduan Manajemen EMR Amazon.

Saat Anda mengaktifkan enkripsi disk lokal menggunakan kunci KMS, proses enkripsi berfungsi seperti ini:

1. Ketika setiap node cluster diluncurkan, ia mengirimkan GenerateDataKeypermintaan keAWS KMS, menentukan ID kunci dari kunci KMS yang Anda pilih ketika Anda mengaktifkan enkripsi disk lokal untuk cluster.
2. AWS KMS menghasilkan kunci enkripsi data yang unik (kunci data) kemudian mengirimkan dua salinan kunci data ini ke simpul. Satu salinan tidak terenkripsi (plaintext), dan salinan lainnya dienkripsi di bawah kunci KMS.
3. Node menggunakan versi base64 yang dikodekan dari kunci data plaintext sebagai kata sandi yang melindungi kunci LUKS. Simpul menyimpan salinan terenkripsi dari kunci data pada volume boot.
4. Jika simpul melakukan boot ulang, simpul yang di-boot ulang mengirimkan kunci data terenkripsi ke AWS KMS dengan permintaan Dekripsi.
5. AWS KMSmendekripsi kunci data terenkripsi menggunakan kunci KMS yang sama yang digunakan untuk mengenkripsinya, dan kemudian mengirimkan kunci data yang didekripsi (plaintext) ke node.
6. Node menggunakan versi base64 yang dikodekan dari kunci data plaintext sebagai kata sandi untuk membuka kunci LUKS.

Konteks enkripsi

Setiap AWS layanan yang terintegrasi dengan AWS KMS dapat menentukan konteks enkripsi ketika layanan menggunakan AWS KMS untuk menghasilkan kunci data atau untuk mengenkripsi atau mendekripsi data. Konteks enkripsi adalah informasi diautentikasi tambahan yang digunakan AWS KMS untuk memeriksa integritas data. Ketika layanan menentukan konteks enkripsi untuk operasi enkripsi, itu harus menentukan konteks enkripsi yang sama untuk operasi dekripsi yang sesuai atau dekripsi akan gagal. Konteks enkripsi juga ditulis ke file AWS CloudTrail log, yang dapat membantu Anda memahami mengapa kunci KMS tertentu digunakan.

Bagian berikut menjelaskan konteks enkripsi yang digunakan dalam setiap skenario enkripsi EMR Amazon yang menggunakan kunci KMS.

Konteks enkripsi untuk enkripsi EMRFS dengan SSE-KMS

Dengan SSE-KMS, cluster EMR Amazon mengirimkan data ke Amazon S3, dan kemudian Amazon S3 menggunakan kunci KMS untuk mengenkripsi data sebelum menyimpannya ke bucket S3. Dalam hal ini, Amazon S3 menggunakan Nama Sumber Daya Amazon (ARN) dari objek S3 sebagai konteks enkripsi dengan masing-masing GenerateDataKeydan Dekripsi permintaan yang dikirimkannya. AWS KMS Contoh berikut menunjukkan representasi JSON dari konteks enkripsi yang digunakan Amazon S3.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Konteks enkripsi untuk enkripsi EMRFS dengan CSE-KMS

Dengan CSE-KMS, cluster EMR Amazon menggunakan kunci KMS untuk mengenkripsi data sebelum mengirimnya ke Amazon S3 untuk penyimpanan. Dalam hal ini, cluster menggunakan Amazon Resource Name (ARN) dari kunci KMS sebagai konteks enkripsi dengan masing-masing GenerateDataKeydan Decrypt request yang dikirimkan. AWS KMS Contoh berikut menunjukkan representasi JSON dari konteks enkripsi yang digunakan klaster.

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Konteks enkripsi untuk enkripsi disk lokal dengan LUKS

Ketika kluster EMR Amazon menggunakan enkripsi disk lokal dengan LUKS, node cluster tidak menentukan konteks enkripsi dengan GenerateDataKeydan Dekripsi permintaan yang mereka kirim. AWS KMS

Bagaimana Nitro Enclaves AWS menggunakan AWS KMS

AWS KMSmendukung pengesahan kriptografi untuk AWS Nitro Enclave. Aplikasi yang mendukung AWS Nitro Enclave memanggil operasi AWS KMS kriptografi berikut dengan dokumen pengesahan yang ditandatangani untuk enclave. AWS KMSAPI ini memverifikasi bahwa dokumen pengesahan berasal dari kantong Nitro. Kemudian, alih-alih mengembalikan data teks biasa dalam respons, API ini mengenkripsi plaintext dengan kunci publik dari dokumen pengesahan dan mengembalikan ciphertext yang hanya dapat didekripsi oleh kunci pribadi yang sesuai di enklave.

- Dekripsi

- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

Tabel berikut menunjukkan bagaimana respons terhadap permintaan enclave Nitro berbeda dari respons standar untuk setiap operasi API.

Operasi AWS KMS	Respon standar	Tanggapan untuk AWS Nitro Enclave
`Decrypt`	Mengembalikan data plaintext	Mengembalikan data plaintext yang dienkripsi oleh kunci publik dari dokumen pengesahan
`GenerateDataKey`	Mengembalikan salinan plaintext dari kunci data (Juga mengembalikan salinan kunci data yang dienkripsi oleh kunci KMS)	Mengembalikan salinan kunci data yang dienkripsi oleh kunci publik dari dokumen pengesahan (Juga mengembalikan salinan kunci data yang dienkripsi oleh kunci KMS)
`GenerateDataKeyPair`	Mengembalikan salinan plaintext dari kunci pribadi (Juga mengembalikan kunci publik dan salinan kunci pribadi yang dienkripsi oleh kunci KMS)	Mengembalikan salinan kunci pribadi yang dienkripsi oleh kunci publik dari dokumen pengesahan (Juga mengembalikan kunci publik dan salinan kunci pribadi yang dienkripsi oleh kunci KMS)
`GenerateRandom`	Mengembalikan string byte acak	Mengembalikan string byte acak dienkripsi oleh

Operasi AWS KMS	Respon standar	Tanggapan untuk AWS Nitro Enclave
		kunci publik dari dokumen pengesahan

AWS KMSmendukung kunci kondisi kebijakan yang dapat Anda gunakan untuk mengizinkan atau menolak operasi enclave dengan AWS KMS kunci berdasarkan konten dokumen pengesahan. Anda juga dapat memantau permintaan AWS KMS untuk enklave Nitro Anda di log Anda. AWS CloudTrail

Topik

- Cara memanggil AWS KMS API untuk enclave Nitro
- Kunci syarat AWS KMS untuk AWS Nitro Enclaves
- Memantau permintaan untuk kantong Nitro

Cara memanggil AWS KMS API untuk enclave Nitro

Untuk memanggil AWS KMS API untuk enclave Nitro, gunakan `Recipient` parameter dalam permintaan untuk menyediakan dokumen pengesahan yang ditandatangani untuk enclave dan algoritma enkripsi untuk digunakan dengan kunci publik enclave. Ketika permintaan menyertakan `Recipient` parameter dengan dokumen pengesahan yang ditandatangani, respons menyertakan `CiphertextForRecipient` bidang dengan ciphertext yang dienkripsi oleh kunci publik. Bidang plaintext adalah nol atau kosong.

`Recipient`Parameter harus menentukan dokumen pengesahan yang ditandatangani dari enklave AWS Nitro. AWS KMSbergantung pada tanda tangan digital untuk dokumen pengesahan enklave untuk membuktikan bahwa kunci publik dalam permintaan berasal dari kantong yang valid. Anda tidak dapat menyediakan sertifikat Anda sendiri untuk menandatangani dokumen pengesahan secara digital.

Untuk menentukan `Recipient` parameter, gunakan AWSNitro Enclave SDK atau SDK apa pun. AWS AWSNitro Enclave SDK, yang didukung hanya dalam enklave Nitro, secara otomatis menambahkan `Recipient` parameter dan nilainya ke setiap permintaan. AWS KMS Untuk membuat permintaan enklaf Nitro di AWS SDK, Anda harus menentukan `Recipient` parameter dan nilainya. Support untuk pengesahan kriptografi enclave Nitro di AWS SDK diperkenalkan pada Maret 2023.

AWS KMSmendukung kunci kondisi kebijakan yang dapat Anda gunakan untuk mengizinkan atau menolak operasi enclave dengan AWS KMS kunci berdasarkan konten dokumen pengesahan. Anda juga dapat memantau permintaan AWS KMS untuk enklave Nitro Anda di log Anda. AWS CloudTrail

Untuk informasi terperinci tentang `Recipient` parameter dan bidang `CiphertextForRecipient` respons AWS, lihat Dekripsi,, dan GenerateRandomtopik di Referensi AWS Key Management Service API GenerateDataKeyGenerateDataKeyPair, AWSNitro Enclave SDK, atau SDK apa pun. AWS Untuk informasi tentang cara menyiapkan data dan kunci data untuk enkripsi, lihat Menggunakan pengesahan kriptografi dengan AWS KMS.

Kunci syarat AWS KMS untuk AWS Nitro Enclaves

Anda dapat menentukan kunci kondisi dalam kebijakan utama dan kebijakan IAM yang mengontrol akses ke AWS KMS sumber daya Anda. Pernyataan kebijakan yang mencakup kunci kondisi hanya efektif jika kondisinya terpenuhi.

AWS KMSmenyediakan kunci kondisi yang membatasi izin untuk Dekripsi,, GenerateDataKeyGenerateDataKeyPair, dan GenerateRandomoperasi berdasarkan isi dokumen pengesahan yang ditandatangani dalam permintaan. Kunci kondisi ini yang hanya berfungsi ketika permintaan untuk AWS KMS operasi menyertakan `Recipient` parameter dengan dokumen pengesahan yang valid dari enklave AWS Nitro. Untuk menentukan `Recipient` parameter, gunakan AWSNitro Enclave SDK atau SDK apa pun. AWS

Kunci AWS KMS kondisi khusus enclave valid dalam pernyataan kebijakan utama dan pernyataan kebijakan IAM meskipun tidak muncul di konsol IAM atau Referensi Otorisasi Layanan IAM.

km:RecipientAttestation: 384 ImageSha

Kunci Syarat AWS KMS	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:ImageSha384`	String	Bernilai tunggal	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
				Kebijakan kunci dan kebijakan IAM

Kunci Syarat AWS KMS	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan
			`GenerateRandom`	

Kunci `kms:RecipientAttestation:ImageSha384` kondisi mengontrol akses ke`Decrypt`,`GenerateDataKey`,`GenerateDataKeyPair`, dan `GenerateRandom` dengan kunci KMS saat gambar digest dari dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan nilai dalam kunci kondisi. `ImageSha384`Nilai sesuai dengan PCR0 dalam dokumen pengesahan. Kunci kondisi ini hanya efektif jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani untuk enclave Nitro. AWS

Nilai ini juga termasuk dalam CloudTrailacara untuk permintaan ke AWS KMS kantong Nitro.

> **Note**
>
> Kunci kondisi ini valid dalam pernyataan kebijakan utama dan pernyataan kebijakan IAM meskipun tidak muncul di konsol IAM atau Referensi Otorisasi Layanan IAM.

Misalnya, pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran untuk menggunakan kunci KMS untuk Dekripsi,, GenerateDataKeyGenerateDataKeyPair, dan operasi. GenerateRandom Kunci `kms:RecipientAttestation:ImageSha384` kondisi memungkinkan operasi hanya jika nilai intisari gambar (PCR0) dari dokumen pengesahan dalam permintaan cocok dengan nilai intisari gambar dalam kondisi. Kunci kondisi ini hanya efektif jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani untuk enclave Nitro. AWS

Jika permintaan tidak menyertakan dokumen pengesahan yang valid dari kantong AWS Nitro, izin ditolak karena kondisi ini tidak terpenuhi.

```
{
  "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
    }
  }
}
```

km ::PCR RecipientAttestation <PCR_ID>

Kunci Syarat AWS KMS	Jenis Syarat	Jenis nilai	Operasi API	Jenis Kebijakan

`kms:RecipientAttestation:PCR<PCR_ID>`	String	Bernilai tunggal	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				Kebijakan kunci dan kebijakan IAM

Kunci `kms:RecipientAttestation:PCR<PCR_ID>` kondisi mengontrol akses ke`Decrypt`,, `GenerateDataKeyGenerateDataKeyPair`, dan `GenerateRandom` dengan kunci KMS hanya jika konfigurasi platform register (PCR) dari dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan PCR dalam kunci kondisi. Kunci kondisi ini hanya efektif jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani dari enklave Nitro. AWS

Nilai ini juga termasuk dalam CloudTrailperistiwa yang mewakili permintaan AWS KMS untuk kantong Nitro.

> **Note**
>
> Kunci kondisi ini valid dalam pernyataan kebijakan utama dan pernyataan kebijakan IAM meskipun tidak muncul di konsol IAM atau Referensi Otorisasi Layanan IAM.

Untuk menentukan nilai PCR, gunakan format berikut. Gabungkan ID PCR ke nama kunci syarat. Nilai PCR harus berupa string heksadesimal huruf kecil hingga 96 byte.

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

Misalnya, kunci kondisi berikut menentukan nilai tertentu untuk PCR1, yang sesuai dengan hash kernel yang digunakan untuk enclave dan proses bootstrap.

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

Contoh pernyataan kebijakan kunci berikut memungkinkan `data-processing` peran untuk menggunakan kunci KMS untuk operasi Dekripsi.

Kunci syarat `kms:RecipientAttestation:PCR` dalam pernyataan ini memungkinkan operasi hanya jika nilai PCR1 dalam dokumen pengesahan yang ditandatangani dalam permintaan cocok dengan nilai `kms:RecipientAttestation:PCR1` dalam syarat. Gunakan operator kebijakan `StringEqualsIgnoreCase` untuk mewajibkan perbandingan nilai PCR yang tidak peka huruf besar/kecil.

Jika permintaan tidak menyertakan dokumen pengesahan, izin ditolak karena kondisi ini tidak terpenuhi.

```
{
  "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":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530082c6f15
    }
  }
}
```

Memantau permintaan untuk kantong Nitro

Anda dapat menggunakan AWS CloudTrail log Anda untuk memantau Dekripsi,, GenerateDataKeyGenerateDataKeyPair, dan GenerateRandomoperasi untuk enklave AWS Nitro. Dalam entri log ini, `additionalEventData` bidang memiliki `recipient` bidang dengan ID modul (`attestationDocumentModuleId`), image digest (`attestationDocumentEnclaveImageDigest`), dan register konfigurasi platform (PCR) dari dokumen pengesahan dalam permintaan. Bidang ini disertakan hanya jika `Recipient` parameter dalam permintaan menentukan dokumen pengesahan yang ditandatangani dari enklave Nitro. AWS

ID modul adalah ID enclave dari enclave Nitro. Intisari gambar adalah hash SHA384 dari gambar enclave. Anda dapat menggunakan intisari gambar dan nilai PCR dalam kondisi untuk kebijakan utama dan kebijakan IAM. Untuk informasi tentang PCR, lihat Tempat mendapatkan pengukuran enclave di Panduan Pengguna AWSNitro Enclave.

Bagian ini menunjukkan contoh entri CloudTrail log untuk setiap permintaan enclave Nitro yang didukung. AWS KMS

Dekripsi (untuk kantong)

Contoh berikut menunjukkan entri AWS CloudTrail log operasi Dekripsi untuk enclave AWS Nitro.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "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 (untuk kantong)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateDataKeyoperasi untuk kantong AWS Nitro.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair (untuk kantong)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateDataKeyPairoperasi untuk kantong AWS Nitro.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
```

```
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom (untuk kantong)

Contoh berikut menunjukkan entri AWS CloudTrail log GenerateRandomoperasi untuk kantong AWS Nitro.

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
```

```
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Bagaimana Amazon Redshift menggunakan AWS KMS

Topik ini membahas bagaimana Amazon Redshift menggunakan AWS KMS untuk mengenkripsi data.

Topik

- Enkripsi Amazon Redshift
- Konteks Enkripsi

Enkripsi Amazon Redshift

Sebuah gudang data Amazon Redshift adalah kumpulan sumber daya komputasi yang disebut simpul, yang diatur ke dalam grup yang disebut klaster. Setiap klaster menjalankan mesin Amazon Redshift dan berisi satu atau lebih database.

Amazon Redshift menggunakan arsitektur berbasis kunci empat tingkat untuk enkripsi. Arsitektur terdiri dari kunci enkripsi data, kunci database, kunci cluster, dan kunci root. Anda dapat menggunakan AWS KMS key sebagai kunci root.

Kunci enkripsi data mengenkripsi blok data dalam klaster. Setiap blok data diberikan kunci AES-256 yang dihasilkan secara acak. Kunci ini dienkripsi dengan menggunakan kunci database untuk klaster.

Kunci database mengenkripsi kunci enkripsi data dalam klaster. Kunci database adalah kunci AES-256 yang dihasilkan secara acak. Ini disimpan pada disk dalam jaringan terpisah dari klaster Amazon Redshift dan diteruskan ke klaster di saluran aman.

Kunci klaster mengenkripsi kunci database untuk klaster Amazon Redshift. Anda dapat menggunakan AWS KMS, AWS CloudHSM, atau modul keamanan perangkat keras eksternal (HSM)

untuk mengelola kunci klaster. Lihat dokumentasi Enkripsi Database Amazon Redshift untuk detail selengkapnya.

Anda dapat meminta enkripsi dengan mencentang kotak yang sesuai di konsol Amazon Redshift. Anda dapat menentukan kunci yang dikelola pelanggan dengan memilih salah satu dari daftar yang muncul di bawah kotak enkripsi. Jika Anda tidak menentukan kunci yang dikelola pelanggan, Amazon Redshift menggunakan Kunci yang dikelola AWSfor Amazon Redshift di bawah akun Anda.

> **Important**
>
> Amazon Redshift hanya mendukung kunci KMS enkripsi simetris. Anda tidak dapat menggunakan kunci KMS asimetris dalam alur kerja enkripsi Amazon Redshift. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Konteks Enkripsi

Setiap layanan yang terintegrasi dengan AWS KMS menentukan konteks enkripsi saat meminta kunci data, mengenkripsi, dan mendekripsi. Konteks enkripsi adalah data yang diautentikasi tambahan (AAD) yang digunakan AWS KMS untuk memeriksa integritas data. Artinya, ketika konteks enkripsi ditentukan untuk operasi enkripsi, layanan juga menentukan untuk operasi dekripsi atau dekripsi tidak akan berhasil. Amazon Redshift menggunakan ID klaster dan waktu penciptaan untuk konteks enkripsi. Di `requestParameters` bidang file CloudTrail log, konteks enkripsi akan terlihat mirip dengan ini.

```
"encryptionContext": {
    "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",
    "aws:redshift:createtime": "20150206T1832Z"
},
```

Anda dapat mencari nama cluster di CloudTrail log Anda untuk memahami operasi apa yang dilakukan dengan menggunakan AWS KMS key (kunci KMS). Operasi termasuk enkripsi klaster, klaster dekripsi, dan menghasilkan kunci data.

Bagaimana Amazon Relational Database Service (Amazon RDS) menggunakan AWS KMS

Anda dapat menggunakan Amazon Relational Database Service (Amazon RDS) untuk menyiapkan, mengoperasikan, dan menskalakan basis data relasional di cloud. Anda dapat mengenkripsi sumber daya Amazon RDS di bawah kunci Kunci yang dikelola AWS atau yang dikelola pelanggan. Amazon RDS dibangun di atas Amazon Elastic Block Store (Amazon EBS) untuk menyediakan enkripsi disk penuh untuk volume database.

Untuk informasi terperinci tentang cara Amazon RDS menggunakan kunci KMS untuk melindungi sumber daya Anda, lihat Mengenkripsi sumber daya Amazon RDS AWS KMS dan manajemen kunci di Panduan Pengguna Amazon RDS.

Bagaimana AWS Secrets Manager menggunakan AWS KMS

AWS Secrets Manager adalah layanan AWS yang mengenkripsi dan menyimpan rahasia Anda, dan mendekripsi secara transparan dan mengembalikannya kepada Anda dalam plaintext. Ini dirancang khusus untuk menyimpan rahasia aplikasi, seperti kredensial masuk, yang berubah secara berkala dan tidak boleh dikode keras atau disimpan dalam plaintext dalam aplikasi. Di tempat kredensial dikode keras atau tabel pencarian, aplikasi Anda memanggil Secrets Manager.

Secrets Manager juga mendukung fitur yang secara berkala memutar rahasia yang terkait dengan database yang umum digunakan. Ini selalu mengenkripsi rahasia diputar baru sebelum disimpan.

Secrets Manager terintegrasi dengan AWS Key Management Service (AWS KMS) untuk mengenkripsi setiap versi dari setiap nilai rahasia dengan kunci data unik yang dilindungi oleh file. AWS KMS key Integrasi ini melindungi rahasia Anda di bawah kunci enkripsi yang tidak pernah meninggalkan AWS KMS tidak terenkripsi. Ini juga memungkinkan Anda untuk mengatur izin khusus pada kunci KMS dan mengaudit operasi yang menghasilkan, mengenkripsi, dan mendekripsi kunci data yang melindungi rahasia Anda.

Untuk informasi tentang cara Secrets Manager menggunakan kunci KMS untuk melindungi rahasia Anda, lihat Mengenkripsi dan mendekripsi rahasia di Panduan Pengguna. AWS Secrets Manager

Bagaimana Amazon Simple Email Service (Amazon SES) menggunakan AWS KMS

Anda dapat menggunakan Amazon Simple Email Service (Amazon SES) untuk menerima email, dan (secara opsional) untuk mengenkripsi pesan email yang diterima sebelum menyimpannya dalam bucket Amazon Simple Storage Service (Amazon S3) yang Anda pilih. Saat Anda mengonfigurasi Amazon SES untuk mengenkripsi pesan email, Anda harus memilih AWS KMS AWS KMS keydi mana Amazon SES mengenkripsi pesan. Anda dapat memilih Kunci yang dikelola AWSuntuk Amazon SES (aliasnya`aaws/ses`), atau Anda dapat memilih kunci terkelola pelanggan simetris yang Anda buat. AWS KMS

> **Important**
>
> Amazon SES hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi pesan email Amazon SES Anda. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Untuk informasi selengkapnya tentang menerima email menggunakan Amazon SES, masuk ke Menerima Email dengan Amazon SES di Panduan Developer Amazon Simple Email Service.

Topik

Gambaran umum enkripsi Amazon SES menggunakan AWS KMS

Bila Anda mengonfigurasi Amazon SES untuk menerima email dan mengenkripsi pesan email sebelum menyimpannya ke bucket S3 Anda, prosesnya akan bekerja seperti ini:

1. Anda membuat aturan tanda terima untuk Amazon SES, menentukan tindakan S3, bucket S3 untuk penyimpanan, dan enkripsi untuk. AWS KMS key
2. Amazon SES menerima pesan email yang sesuai dengan aturan penerimaan Anda.

3. Amazon SES meminta kunci data unik yang dienkripsi dengan kunci KMS yang Anda tentukan dalam aturan tanda terima yang berlaku.
4. AWS KMSmembuat kunci data baru, mengenkripsi dengan kunci KMS yang ditentukan, dan kemudian mengirimkan salinan kunci data terenkripsi dan plaintext ke Amazon SES.
5. Amazon SES menggunakan kunci data plaintext untuk mengenkripsi email kemudian menghapus kunci data plaintext dari memori sesegera mungkin setelah digunakan.
6. Amazon SES menempatkan pesan email terenkripsi dan kunci data terenkripsi dalam bucket S3 yang ditentukan. Kunci data terenkripsi disimpan sebagai metadata dengan pesan email terenkripsi.

Untuk mencapai Step 3 melalui Step 6, Amazon SES menggunakan klien enkripsi Amazon S3 yang disediakan AWS. Gunakan klien yang sama untuk mengambil pesan email terenkripsi dari Amazon S3 dan mendekripsinya. Untuk informasi selengkapnya, lihat Mendapatkan dan mendekripsi pesan email.

Konteks enkripsi Amazon SES

Ketika Amazon SES meminta kunci data untuk mengenkripsi pesan email yang Anda terima (Step 3 di Gambaran umum enkripsi Amazon SES menggunakan AWS KMS), itu termasuk konteks enkripsi dalam permintaan. Konteks enkripsi memberikan data yang diautentikasi tambahan (AAD) yang digunakan AWS KMS untuk memastikan integritas data. Konteks enkripsi juga ditulis ke file AWS CloudTrail log Anda, yang dapat membantu Anda memahami mengapa diberikan AWS KMS key (kunci KMS) digunakan. Amazon SES menggunakan konteks enkripsi berikut:

- ID dari Akun AWS di mana Anda telah mengonfigurasi Amazon SES untuk menerima pesan email
- Nama aturan dari aturan penerimaan Amazon SES yang memanggil tindakan S3 pada pesan email
- ID pesan Amazon SES untuk pesan email

Contoh berikut menunjukkan representasi JSON dari konteks enkripsi yang digunakan Amazon SES:

```
{
  "aws:ses:source-account": "111122223333",
  "aws:ses:rule-name": "example-receipt-rule-name",
  "aws:ses:message-id": "d6iitobk75ur44p8kdnnp7g2n800"
}
```

Memberikan izin Amazon SES untuk menggunakan AWS KMS key

Untuk mengenkripsi pesan email Anda, Anda dapat menggunakan Kunci yang dikelola AWSdi akun Anda untuk Amazon SES (`aws/ses`), atau Anda dapat menggunakan kunci terkelola pelanggan yang Anda buat. Amazon SES sudah memiliki izin untuk menggunakan Kunci yang dikelola AWS atas nama Anda. Namun, jika Anda menentukan kunci terkelola pelanggan saat menambahkan tindakan S3 ke aturan tanda terima Amazon SES, Anda harus memberi izin kepada Amazon SES untuk menggunakan kunci KMS untuk mengenkripsi pesan email Anda.

Untuk memberikan izin kepada Amazon SES untuk menggunakan kunci terkelola pelanggan Anda, tambahkan pernyataan berikut ke kebijakan kunci KMS tersebut:

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

Ganti *`ACCOUNT-ID-WITHOUT-HYPHENS`* dengan ID 12 digit Akun AWS tempat Anda mengonfigurasi Amazon SES untuk menerima pesan email. Pernyataan kebijakan ini memungkinkan Amazon SES untuk mengenkripsi data dengan kunci KMS ini hanya dalam kondisi berikut:

- Amazon SES harus menentukan `aws:ses:rule-name` dan `aws:ses:message-id` di `EncryptionContext` dari permintaan API AWS KMS mereka.
- Amazon SES harus menentukan `aws:ses:source-account` di `EncryptionContext` dari permintaan API AWS KMS mereka, dan nilai untuk `aws:ses:source-account` harus sesuai dengan ID Akun AWS yang ditentukan dalam kebijakan kunci.

Untuk informasi selengkapnya tentang konteks enkripsi yang digunakan Amazon SES saat mengenkripsi pesan email Anda, lihat Konteks enkripsi Amazon SES. Untuk informasi umum tentang bagaimana AWS KMS menggunakan konteks enkripsi, lihat konteks enkripsi.

Mendapatkan dan mendekripsi pesan email

Amazon SES tidak memiliki izin untuk mendekripsi pesan email terenkripsi Anda dan tidak dapat mendekripsinya untuk Anda. Anda harus menulis kode untuk mendapatkan pesan email Anda dari Amazon S3 dan mendekripsinya. Untuk membuatnya lebih mudah, gunakan klien enkripsi Amazon S3. SDK AWS berikut termasuk klien enkripsi Amazon S3:

- AWS SDK for Java— Lihat AmazonS3EncryptionClientdan AmazonS3EncryptionClientV2di Referensi AWS SDK for Java API.
- AWS SDK for Ruby — Lihat Aws::S3::Encryption::Client di Referensi API AWS SDK for Ruby.
- AWS SDK for .NET — Lihat AmazonS3EncryptionClient di Referensi API AWS SDK for .NET.
- AWS SDK for Go — Lihat s3crypto di Referensi API AWS SDK for Go.

Klien enkripsi Amazon S3 menyederhanakan pekerjaan membangun permintaan yang diperlukan ke Amazon S3 untuk mengambil pesan email terenkripsi dan AWS KMS untuk mendekripsi kunci data terenkripsi pesan, dan mendekripsi pesan e-mail. Misalnya, untuk berhasil mendekripsi kunci data terenkripsi Anda harus melewati konteks enkripsi yang sama yang dilewati Amazon SES saat meminta kunci data dari AWS KMS (Step 3 di Gambaran umum enkripsi Amazon SES menggunakan AWS KMS). Klien enkripsi Amazon S3 menangani ini, dan banyak pekerjaan lainnya, untuk Anda.

Untuk kode contoh yang menggunakan klien enkripsi Amazon S3 di AWS SDK for Java untuk melakukan dekripsi sisi klien, lihat berikut ini:

- Menggunakan kunci KMS yang disimpan AWS KMS dalam Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Enkripsi Amazon S3 dengan AWS Key Management Service pada Blog Developer AWS.

Bagaimana Amazon Simple Storage Service (Amazon S3) menggunakan AWS KMS

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menyimpan data sebagai objek di dalam bucket. Bucket dan objek di dalamnya bersifat pribadi dan hanya dapat diakses jika Anda secara eksplisit memberikan izin akses.

Amazon S3 terintegrasi dengan AWS Key Management Service (AWS KMS) untuk menyediakan enkripsi sisi server objek Amazon S3. Amazon S3 menggunakan AWS KMS kunci untuk mengenkripsi objek Amazon S3 Anda. Kunci enkripsi yang melindungi objek Anda tidak pernah dibiarkan tidak AWS KMS terenkripsi. Integrasi ini juga memungkinkan Anda untuk mengatur izin pada AWS KMS kunci dan mengaudit operasi yang menghasilkan, mengenkripsi, dan mendekripsi kunci data yang melindungi rahasia Anda.

Untuk mengurangi volume panggilan Amazon S3AWS KMS, gunakan kunci bucket Amazon S3, yang key-encryption-keys dilindungi kunci KMS yang digunakan kembali untuk waktu terbatas dalam Amazon S3. Kunci bucket dapat mengurangi biaya AWS KMS permintaan hingga 99 persen. Anda dapat mengonfigurasi kunci bucket untuk semua objek di bucket Amazon S3, atau untuk objek tertentu di bucket Amazon S3.

Untuk informasi selengkapnya tentang cara Amazon S3 menggunakanAWS KMS, lihat Melindungi data menggunakan enkripsi sisi server dengan kunci KMS (SSE-KMS) di Panduan Pengguna Amazon S3.

Bagaimana Parameter Store AWS Systems Manager menggunakan AWS KMS

Dengan Parameter Store AWS Systems Manager, Anda dapat membuat parameter string yang aman, yang merupakan parameter yang memiliki nama parameter plaintext dan nilai parameter terenkripsi. Pelajari bagaimana Parameter Store menggunakan AWS KMS untuk mengenkripsi nilai parameter string yang aman.

Dengan Parameter Store Anda dapat membuat, menyimpan, dan mengelola data sebagai parameter dengan nilai-nilai. Anda dapat membuat parameter di Parameter Store dan menggunakannya dalam beberapa aplikasi dan layanan tunduk pada kebijakan dan izin yang Anda desain. Ketika Anda perlu mengubah nilai parameter, Anda mengubah satu instans, daripada mengelola perubahan

rawan kesalahan ke berbagai sumber. Parameter Store mendukung struktur hierarkis untuk nama parameter, sehingga Anda dapat memenuhi syarat parameter untuk penggunaan tertentu.

Untuk mengelola data sensitif, Anda dapat membuat parameter string yang aman. Parameter Store menggunakan AWS KMS keys untuk mengenkripsi nilai parameter parameter string aman saat Anda membuat atau mengubahnya. Ini juga menggunakan kunci KMS untuk mendekripsi nilai parameter saat Anda mengaksesnya. Anda dapat menggunakan Parameter Store Kunci yang dikelola AWSyang dibuat untuk akun Anda atau menentukan kunci yang dikelola pelanggan Anda sendiri.

> **Important**
>
> Parameter Store hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi parameter Anda. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Parameter Store mendukung dua tingkat parameter string yang aman: standar dan tingkat lanjut. Parameter standar, yang tidak dapat melebihi 4096 byte, dienkripsi dan didekripsi langsung di bawah kunci KMS yang Anda tentukan. Untuk mengenkripsi dan mendekripsi parameter string aman tingkat lanjut, Parameter Store menggunakan enkripsi amplop dengan AWS Encryption SDK. Anda dapat mengonversi parameter string aman standar untuk parameter lanjutan, tetapi Anda tidak dapat mengonversi parameter lanjutan menjadi standar. Untuk informasi selengkapnya tentang perbedaan antara parameter string aman standar dan lanjutan, lihat Tentang Parameter Lanjutan Systems Manager dalam Panduan Pengguna AWS Systems Manager.

Topik

- Melindungi parameter string aman standar
- Melindungi parameter string aman tingkat lanjut
- Menetapkan izin untuk mengenkripsi dan mendekripsi nilai parameter
- Konteks enkripsi Parameter Store
- Memecahkan masalah utama KMS di Parameter Store

Melindungi parameter string aman standar

Parameter Store tidak melakukan operasi kriptografis apa pun. Sebaliknya, itu bergantung pada AWS KMS untuk mengenkripsi dan mendekripsi nilai parameter string aman. Ketika Anda membuat

atau mengubah nilai parameter string aman standar, Parameter Store memanggil operasi AWS KMS Enkripsi. Operasi ini menggunakan kunci KMS enkripsi simetris secara langsung untuk mengenkripsi nilai parameter alih-alih menggunakan kunci KMS untuk menghasilkan kunci data.

Anda dapat memilih kunci KMS yang digunakan Parameter Store untuk mengenkripsi nilai parameter. Jika Anda tidak menentukan kunci KMS, Parameter Store menggunakan Systems Manager Kunci yang dikelola AWS yang dibuat secara otomatis di akun Anda. Kunci KMS ini memiliki `aws/ssm` alias.

Untuk melihat kunci `aws/ssm` KMS default untuk akun Anda, gunakan DescribeKeyoperasi di AWS KMS API. Contoh berikut menggunakan perintah `describe-key` di AWS Command Line Interface (AWS CLI) dengan nama alias `aws/ssm`.

```
aws kms describe-key --key-id alias/aws/ssm
```

Untuk membuat parameter string aman standar, gunakan PutParameteroperasi di Systems Manager API. Abaikan parameter `Tier` atau tentukan nilai `Standard`, yang merupakan default. Sertakan parameter `Type` dengan nilai dari `SecureString`. Untuk menentukan kunci KMS, gunakan `KeyId` parameter. Defaultnya adalah Kunci yang dikelola AWS untuk akun Anda`aws/ssm`.

Parameter Store kemudian memanggil AWS KMS `Encrypt` operasi dengan kunci KMS dan nilai parameter plaintext. AWS KMSmengembalikan nilai parameter terenkripsi, yang disimpan Parameter Store dengan nama parameter.

Contoh berikut menggunakan perintah put-parameter Systems Manager dan parameter `--type` dalam AWS CLI untuk membuat parameter string aman. Karena perintah menghilangkan opsional `--tier` dan `--key-id` parameter, Parameter Store membuat parameter string aman standar dan mengenkripsi di bawah Kunci yang dikelola AWS

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString
```

Contoh serupa berikut menggunakan `--key-id` parameter untuk menentukan kunci yang dikelola pelanggan. Contoh menggunakan ID kunci KMS untuk mengidentifikasi kunci KMS, tetapi Anda dapat menggunakan pengidentifikasi kunci KMS yang valid. Karena perintah mengabaikan parameter `Tier` (`--tier`), Parameter Store membuat parameter string aman standar, bukan yang lanjutan.

```
aws ssm put-parameter --name param1 --value "secret" --type SecureString --key-id
 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ketika Anda mendapatkan parameter string aman dari Parameter Store, nilainya dienkripsi. Untuk mendapatkan parameter, gunakan GetParameter operasi di Systems Manager API.

Contoh berikut menggunakan parameter get-parameter Systems Manager di AWS CLI untuk mendapatkan parameter `MyParameter` dari Parameter Store tanpa mendekripsi nilainya.

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

Untuk mendekripsi nilai parameter sebelum mengembalikannya, atur parameter `WithDecryption` dari `GetParameter` ke `true`. Saat Anda menggunakan `WithDecryption`, Parameter Store memanggil operasi AWS KMS Dekripsi atas nama Anda untuk mendekripsi nilai parameter. Akibatnya, permintaan `GetParameter` akan mengembalikan parameter dengan nilai parameter plaintext, seperti yang ditunjukkan dalam contoh berikut.

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

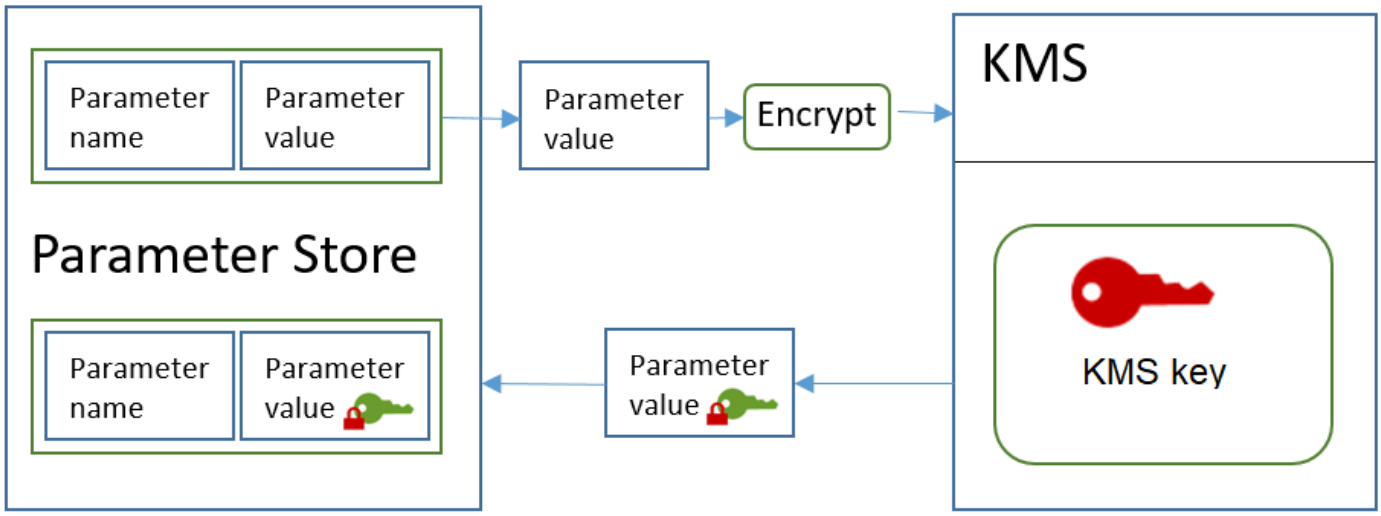
Alur kerja berikut menunjukkan bagaimana Parameter Store menggunakan kunci KMS untuk mengenkripsi dan mendekripsi parameter string aman standar.

Mengenkripsi parameter standar

1. Saat Anda menggunakan `PutParameter` untuk membuat parameter string aman, Parameter Store mengirimkan permintaan `Encrypt` untuk AWS KMS. Permintaan tersebut mencakup nilai parameter plaintext, kunci KMS yang Anda pilih, dan konteks enkripsi Parameter Store. Selama

transmisi ke AWS KMS, nilai plaintext dalam parameter string aman dilindungi oleh Transport Layer Security (TLS).

2. AWS KMSmengenkripsi nilai parameter dengan kunci KMS dan konteks enkripsi yang ditentukan. Ini mengembalikan ciphertext ke Parameter Store, yang menyimpan nama parameter dan nilai terenkripsi.



Dekripsi parameter standar

1. Ketika Anda menyertakan parameter `WithDecryption` dalam permintaan `GetParameter`, Parameter Store mengirimkan permintaan `Decrypt` untuk AWS KMS dengan nilai parameter string aman terenkripsi dan Konteks enkripsi Parameter Store.
2. AWS KMSmenggunakan kunci KMS yang sama dan konteks enkripsi yang disediakan untuk mendekripsi nilai terenkripsi. Ini mengembalikan plaintext (didekripsi) nilai parameter ke Parameter Store. Selama transmisi, data plaintext dilindungi oleh TLS.
3. Parameter Store mengembalikan nilai parameter plaintext untuk Anda dalam respons `GetParameter`.

Melindungi parameter string aman tingkat lanjut

Saat Anda menggunakan `PutParameter` untuk membuat parameter string aman lanjutan, Parameter Store menggunakan enkripsi amplop dengan AWS Encryption SDK dan enkripsi simetris AWS KMS key untuk melindungi nilai parameter. Setiap nilai parameter lanjutan dienkripsi di bawah kunci data unik, dan kunci data dienkripsi di bawah kunci KMS. Anda dapat menggunakan Kunci yang dikelola AWSfor the account (`aws/ssm`) atau kunci yang dikelola pelanggan.

AWS Encryption SDK adalah pustaka sisi klien sumber terbuka yang membantu Anda mengenkripsi dan mendekripsi data menggunakan standar industri dan praktik terbaik. Ini didukung pada beberapa platform dan dalam beberapa bahasa pemrograman, termasuk antarmuka baris perintah. Anda dapat melihat kode sumber dan berkontribusi pada pengembangannya di GitHub.

Untuk setiap nilai parameter string aman, Parameter Store memanggil AWS Encryption SDK untuk mengenkripsi nilai parameter menggunakan kunci data unik yang AWS KMS menghasilkan (GenerateDataKey). AWS Encryption SDK mengembalikan ke Parameter Store sebuah pesan terenkripsi yang mencakup nilai parameter terenkripsi dan salinan terenkripsi dari kunci data unik. Parameter Store menyimpan seluruh pesan terenkripsi dalam nilai parameter string aman. Kemudian, ketika Anda mendapatkan nilai parameter string aman lanjutan, Parameter Store menggunakan AWS Encryption SDK untuk mendekripsi nilai parameter. Ini memerlukan panggilan ke AWS KMS untuk mendekripsi kunci data terenkripsi.

Untuk membuat parameter string aman lanjutan, gunakan PutParameteroperasi di Systems Manager API. Atur nilai parameter `Tier` ke `Advanced`. Sertakan parameter `Type` dengan nilai dari `SecureString`. Untuk menentukan kunci KMS, gunakan `KeyId` parameter. Defaultnya adalah Kunci yang dikelola AWS untuk akun Anda`aws/ssm`.

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString --
tier Advanced
```

Contoh serupa berikut menggunakan `--key-id` parameter untuk menentukan kunci yang dikelola pelanggan. Contoh menggunakan Nama Sumber Daya Amazon (ARN) dari kunci KMS, tetapi Anda dapat menggunakan pengidentifikasi kunci KMS yang valid.

```
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
```

Ketika Anda mendapatkan parameter string aman dari Parameter Store, nilainya adalah pesan terenkripsi yang dikembalikan AWS Encryption SDK. Untuk mendapatkan parameter, gunakan GetParameter operasi di Systems Manager API.

Contoh berikut menggunakan operasi `GetParameter` Systems Manager untuk mendapatkan parameter `MyParameter` dari Parameter Store tanpa mendekripsi nilainya.

```
$  aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

Untuk mendekripsi nilai parameter sebelum mengembalikannya, atur parameter `WithDecryption` dari `GetParameter` ke `true`. Saat Anda menggunakan `WithDecryption`, Parameter Store memanggil operasi AWS KMS Dekripsi atas nama Anda untuk mendekripsi nilai parameter. Akibatnya, permintaan `GetParameter` akan mengembalikan parameter dengan nilai parameter plaintext, seperti yang ditunjukkan dalam contoh berikut.

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Anda tidak dapat mengonversi parameter string aman lanjutan ke parameter standar, tetapi Anda dapat mengonversi string aman standar ke lanjutan. Untuk mengonversi parameter string aman standar ke string aman lanjutan, gunakan operasi `PutParameter` dengan parameter `Overwrite`. `Type` harus `SecureString` dan nilai `Tier` harus `Advanced`. `KeyId`Parameter, yang mengidentifikasi kunci yang dikelola pelanggan, adalah opsional. Jika Anda menghilangkannya, Parameter Store menggunakan Kunci yang dikelola AWS untuk akun. Anda dapat menentukan kunci KMS yang memiliki izin untuk digunakan oleh prinsipal, bahkan jika Anda menggunakan kunci KMS yang berbeda untuk mengenkripsi parameter standar.

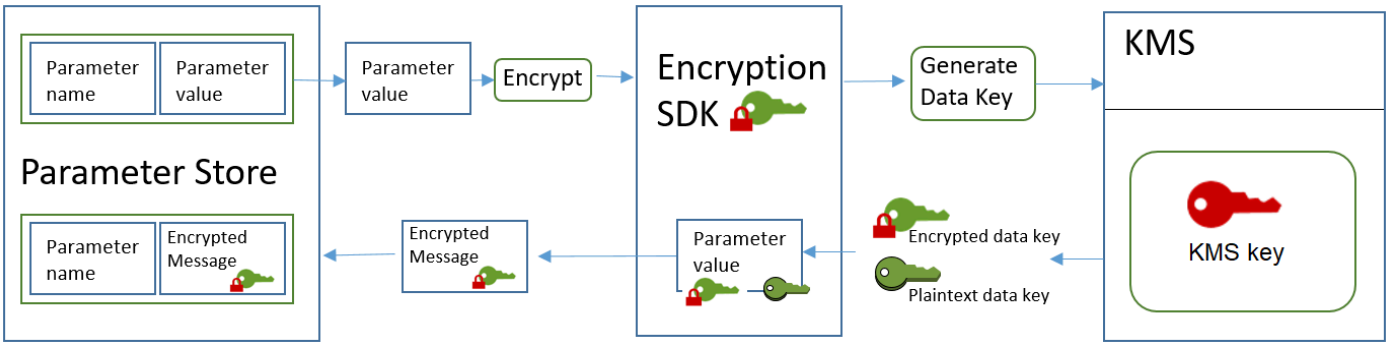
Saat Anda menggunakan parameter `Overwrite`, Parameter Store menggunakan AWS Encryption SDK untuk mengenkripsi nilai parameter. Kemudian ini menyimpan pesan dienkripsi baru di Parameter Store.

```
$ aws ssm put-parameter --name myStdParameter --value "secret_value"  --type
 SecureString --tier Advanced --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --overwrite
```

Alur kerja berikut menunjukkan bagaimana Parameter Store menggunakan kunci KMS untuk mengenkripsi dan mendekripsi parameter string aman lanjutan.

Mengenkripsi parameter lanjutan

1. Saat Anda menggunakan `PutParameter` untuk membuat parameter string aman tingkat lanjut, Parameter Store menggunakan AWS Encryption SDK dan AWS KMS untuk mengenkripsi nilai parameter. Parameter Store memanggil AWS Encryption SDK dengan nilai parameter, kunci KMS yang Anda tentukan, dan konteks enkripsi Parameter Store.
2. AWS Encryption SDKMengirim GenerateDataKeypermintaan AWS KMS dengan pengenal kunci KMS yang Anda tentukan dan konteks enkripsi Parameter Store. AWS KMSmengembalikan dua salinan kunci data unik: satu di plaintext dan satu dienkripsi di bawah kunci KMS. (Konteks enkripsi digunakan saat mengenkripsi kunci data.)
3. AWS Encryption SDK menggunakan kunci data plaintext untuk mengenkripsi nilai parameter. Ini mengembalikan sebuah pesan terenkripsi yang mencakup nilai parameter terenkripsi, kunci data terenkripsi, dan data lainnya, termasuk konteks enkripsi Parameter Store.
4. Parameter Store menyimpan pesan terenkripsi sebagai nilai parameter.



Dekripsi parameter lanjutan

1. Anda dapat menyertakan parameter `WithDecryption` dalam permintaan `GetParameter` untuk mendapatkan parameter string aman lanjutan. Ketika Anda melakukannya, Parameter Store menyampaikan pesan terenkripsi dari nilai parameter ke metode dekripsi AWS Encryption SDK.
2. AWS Encryption SDK memanggil operasi AWS KMS Dekripsi. Ini menyampaikan dalam kunci data terenkripsi dan konteks enkripsi Parameter Store dari pesan terenkripsi.
3. AWS KMSmenggunakan kunci KMS dan konteks enkripsi Parameter Store untuk mendekripsi kunci data terenkripsi. Kemudian ini mengembalikan plaintext (didekripsi) kunci data ke AWS Encryption SDK.

4. AWS Encryption SDK menggunakan kunci data plaintext untuk mendekripsi nilai parameter. Ini mengembalikan plaintext nilai parameter ke Parameter Store.
5. Parameter Store memverifikasi konteks enkripsi dan mengembalikan nilai parameter plaintext untuk Anda dalam respons `GetParameter`.

Menetapkan izin untuk mengenkripsi dan mendekripsi nilai parameter

Untuk mengenkripsi nilai parameter string aman standar, pengguna membutuhkan izin `kms:Encrypt`. Untuk mengenkripsi nilai parameter string aman lanjutan, pengguna membutuhkan izin `kms:GenerateDataKey`. Untuk mendekripsi jenis nilai parameter string aman mana pun, pengguna membutuhkan izin `kms:Decrypt`.

Anda dapat menggunakan kebijakan IAM untuk mengizinkan atau menolak izin bagi pengguna untuk memanggil operasi `PutParameter` dan `GetParameter` Systems Manager.

Jika Anda menggunakan kunci terkelola pelanggan untuk mengenkripsi nilai parameter string aman Anda, Anda dapat menggunakan kebijakan IAM dan kebijakan kunci untuk mengelola enkripsi dan mendekripsi izin. Namun, Anda tidak dapat membuat kebijakan kontrol akses untuk kunci `aws/ssm` KMS default. Untuk informasi rinci tentang mengontrol akses ke kunci yang dikelola pelanggan, lihatKontrol autentikasi dan akses untuk AWS KMS.

Contoh berikut menunjukkan kebijakan IAM yang dirancang untuk parameter string aman standar. Ini memungkinkan pengguna untuk memanggil operasi `PutParameter` Systems Manager pada semua parameter dalam jalur `FinancialParameters`. Kebijakan ini juga memungkinkan pengguna untuk memanggil AWS KMS `Encrypt` operasi pada contoh kunci yang dikelola pelanggan.

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

Contoh selanjutnya menunjukkan kebijakan IAM yang dirancang untuk parameter string aman lanjutan. Ini memungkinkan pengguna untuk memanggil operasi `PutParameter` Systems Manager pada semua parameter dalam jalur `ReservedParameters`. Kebijakan ini juga memungkinkan pengguna untuk memanggil AWS KMS `GenerateDataKey` operasi pada contoh kunci yang dikelola pelanggan.

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

Contoh terakhir juga menunjukkan kebijakan IAM yang dapat digunakan untuk parameter string aman standar atau lanjutan. Ini memungkinkan pengguna untuk memanggil operasi `GetParameter` Systems Manager (dan operasi terkait) pada semua parameter dalam jalur `ITParameters`. Kebijakan ini juga memungkinkan pengguna untuk memanggil AWS KMS `Decrypt` operasi pada contoh kunci yang dikelola pelanggan.

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

Konteks enkripsi Parameter Store

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS secara kriptografi mengikat konteks enkripsi untuk data terenkripsi tersebut. Untuk mendekripsi data, Anda harus lulus dalam konteks enkripsi yang sama.

Anda juga dapat menggunakan konteks enkripsi untuk mengidentifikasi operasi kriptografi dalam audit catatan dan log. Konteks enkripsi muncul di plaintext dalam log, seperti log AWS CloudTrail.

AWS Encryption SDK juga mengambil konteks enkripsi, meskipun ditangani secara berbeda. Parameter Store menyediakan konteks enkripsi untuk metode enkripsi. AWS Encryption SDK secara kriptografi mengikat konteks enkripsi untuk data terenkripsi. Ini juga mencakup konteks enkripsi dalam teks polos di header pesan terenkripsi yang dikembalikan. Namun, tidak seperti AWS KMS, metode dekripsi AWS Encryption SDK tidak mengambil konteks enkripsi sebagai input. Sebaliknya, ketika mendekripsi data, AWS Encryption SDK mendapatkan konteks enkripsi dari pesan terenkripsi. Parameter Store memverifikasi konteks enkripsi menyertakan nilai yang diharapkan sebelum mengembalikan nilai parameter plaintext kepada Anda.

Parameter Store menggunakan konteks enkripsi berikut dalam operasi kriptografinya:

- Kunci: `PARAMETER_ARN`
- Nilai: Amazon Resource Name (ARN) dari parameter yang sedang dienkripsi.

Format konteks penyulitan adalah seperti berikut:

```
"PARAMETER_ARN":"arn:aws:ssm:<REGION_NAME>:<ACCOUNT_ID>:parameter/<parameter-name>"
```

Sebagai contoh, Parameter Store menyertakan konteks enkripsi ini dalam panggilan untuk mengenkripsi dan mendekripsi parameter `MyParameter` dalam Akun AWS dan wilayah contoh.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
```

Jika parameter ada di jalur hierarkis Parameter Store, jalur dan nama termasuk dalam konteks enkripsi. Sebagai contoh, konteks enkripsi ini digunakan ketika mengenkripsi dan mendekripsi parameter `MyParameter` dalam jalur `/ReadableParameters` dalam Akun AWS dan wilayah contoh.

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

Anda dapat mendekripsi nilai parameter string aman terenkripsi dengan memanggil operasi AWS KMS `Decrypt` dengan konteks enkripsi yang benar dan nilai parameter terenkripsi yang dikembalikan operasi `GetParameter` Systems Manager. Namun, kami mendorong Anda untuk mendekripsi nilai parameter Parameter Store dengan menggunakan operasi `GetParameter` dengan parameter `WithDecryption`.

Anda juga dapat menyertakan konteks enkripsi dalam kebijakan IAM. Misalnya, Anda dapat mengizinkan pengguna untuk mendekripsi hanya satu nilai parameter tertentu atau set nilai parameter.

Contoh berikut pernyataan kebijakan IAM memungkinkan pengguna untuk mendapatkan nilai `MyParameter` parameter dan untuk mendekripsi nilainya menggunakan kunci KMS tertentu. Namun izin hanya berlaku ketika konteks enkripsi cocok dengan string tertentu. Izin ini tidak berlaku untuk parameter lain atau kunci KMS, dan panggilan `GetParameter` gagal jika konteks enkripsi tidak cocok dengan string.

Sebelum menggunakan pernyataan kebijakan seperti ini, ganti contoh ARN dengan nilai yang valid.

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

Memecahkan masalah utama KMS di Parameter Store

Untuk melakukan operasi apa pun pada parameter string aman, Parameter Store harus dapat menggunakan kunci AWS KMS KMS yang Anda tentukan untuk operasi yang Anda inginkan. Sebagian besar kegagalan Parameter Store yang terkait dengan kunci KMS disebabkan oleh masalah berikut:

- Kredensi yang digunakan aplikasi tidak memiliki izin untuk melakukan tindakan yang ditentukan pada kunci KMS.

  Untuk memperbaiki kesalahan ini, jalankan aplikasi dengan kredensial yang berbeda atau merevisi IAM atau kebijakan kunci yang mencegah operasi. Untuk bantuan dengan IAM AWS KMS dan kebijakan kunci, lihat Kontrol autentikasi dan akses untuk AWS KMS.
- Kunci KMS tidak ditemukan.

Ini biasanya terjadi ketika Anda menggunakan pengenal yang salah untuk kunci KMS. Temukan pengidentifikasi yang benar untuk kunci KMS dan coba perintahnya lagi.

- Kunci KMS tidak diaktifkan. Ketika ini terjadi, Parameter Store mengembalikan InvalidKeyIdpengecualian dengan pesan kesalahan terperinci dariAWS KMS. Jika status kunci KMS adalah`Disabled`, aktifkan. Jika status `Pending Import`, selesaikan prosedur impor. Jika status kuncinya`Pending Deletion`, batalkan penghapusan kunci atau gunakan kunci KMS yang berbeda.

  Untuk menemukan status kunci kunci KMS di AWS KMS konsol, pada kunci atau Kunci yang dikelola AWShalaman yang dikelola Pelanggan, lihat kolom Status. Untuk menggunakan AWS KMS API untuk menemukan status kunci KMS, gunakan DescribeKeyoperasi.

Bagaimana Amazon WorkMail menggunakan AWS KMS

Topik ini membahas bagaimana Amazon WorkMail menggunakan AWS KMS untuk mengenkripsi pesan email.

Topik

- WorkMail Ikhtisar Amazon
- WorkMail Enkripsi Amazon
- Mengotorisasi penggunaan kunci KMS
- Konteks WorkMail enkripsi Amazon
- Memantau WorkMail interaksi Amazon dengan AWS KMS

WorkMail Ikhtisar Amazon

Amazon WorkMail adalah layanan email dan kalender bisnis yang aman dan terkelola dengan dukungan untuk klien email desktop dan seluler yang ada. Anda dapat membuat WorkMail organisasi Amazon dan menetapkan satu atau beberapa domain email yang Anda miliki. Kemudian Anda dapat membuat kotak pesan untuk pengguna e-mail dan grup distribusi di organisasi.

Amazon WorkMail secara transparan mengenkripsi semua pesan di kotak surat semua WorkMail organisasi Amazon sebelum pesan ditulis ke disk dan mendekripsi pesan secara transparan saat pengguna mengaksesnya. Tidak ada pilihan untuk menonaktifkan enkripsi. Untuk melindungi kunci

enkripsi yang melindungi pesan, Amazon WorkMail terintegrasi dengan AWS Key Management Service (AWS KMS).

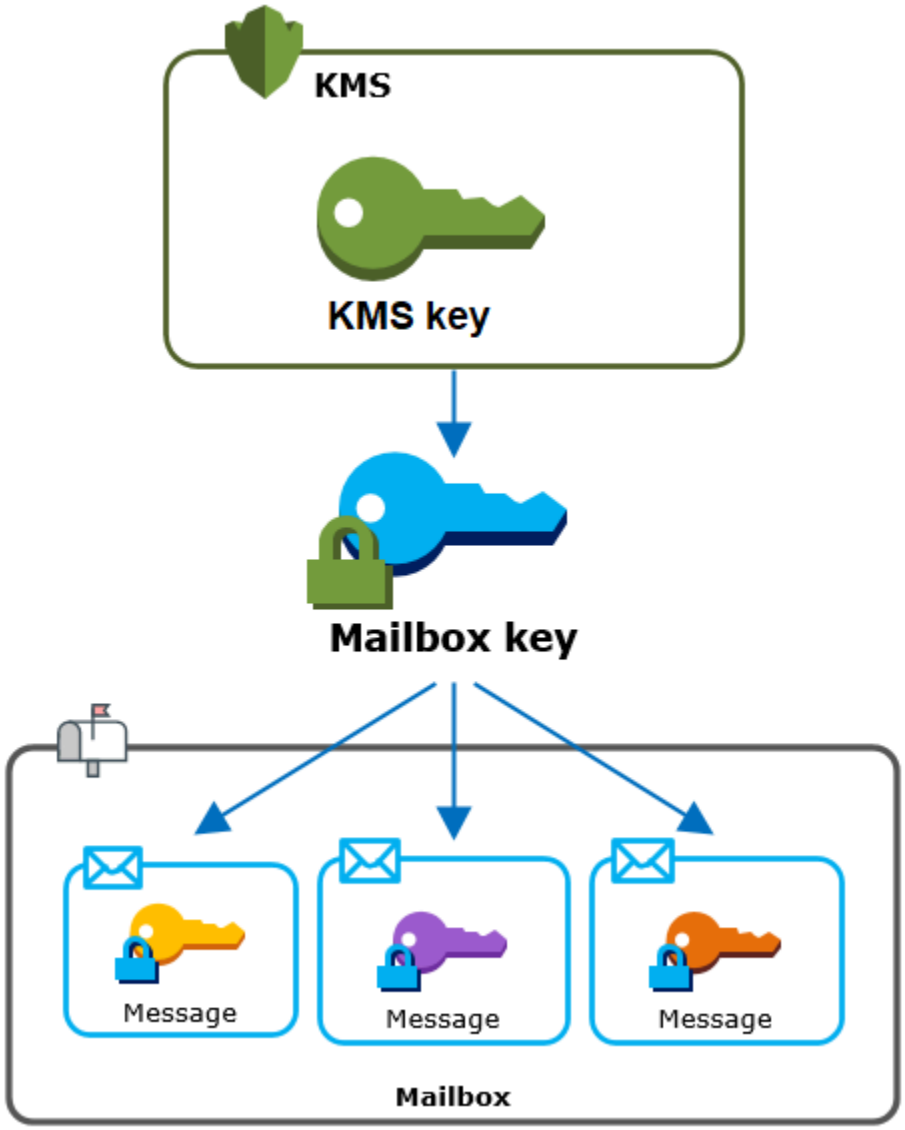
Amazon WorkMail juga menyediakan opsi untuk memungkinkan pengguna mengirim email yang ditandatangani atau dienkripsi. Fitur enkripsi ini tidak menggunakan AWS KMS.

WorkMail Enkripsi Amazon

Di Amazon WorkMail, setiap organisasi dapat berisi beberapa kotak pesan, satu untuk setiap pengguna di organisasi. Semua pesan, termasuk item email dan kalender, disimpan di kotak pesan pengguna.

Untuk melindungi konten kotak pesan di WorkMail organisasi Amazon Anda, Amazon WorkMail mengenkripsi semua pesan kotak pesan sebelum ditulis ke disk. Tidak ada informasi yang disediakan pelanggan disimpan dalam plaintext.

Setiap pesan dienkripsi di bawah kunci enkripsi data yang unik. Kunci pesan dilindungi oleh kunci kotak pesan, yang merupakan kunci enkripsi unik yang hanya digunakan untuk kotak pesan tersebut. Kunci kotak pesan dienkripsi di bawah AWS KMS key untuk organisasi yang tidak pernah dibiarkan tidak terenkripsi. AWS KMS Diagram berikut menunjukkan hubungan pesan terenkripsi, kunci pesan terenkripsi, kunci kotak pesan terenkripsi, dan kunci KMS untuk organisasi. AWS KMS



Kunci KMS untuk organisasi

Saat membuat WorkMail organisasi Amazon, Anda dapat memilih AWS KMS key untuk organisasi tersebut. Kunci KMS ini melindungi semua kunci kotak pesan di organisasi tersebut.

Jika Anda menggunakan prosedur Quick Setup untuk membuat organisasi, Amazon WorkMail menggunakan Kunci yang dikelola AWSfor Amazon WorkMail (`aws/workmail`) di organisasi AndaAkun AWS. Jika Anda menggunakan Pengaturan Standar, Anda dapat memilih Kunci yang dikelola AWS untuk Amazon WorkMail atau kunci yang dikelola pelanggan yang Anda miliki dan kelola. Anda dapat memilih kunci KMS yang sama atau kunci KMS yang berbeda untuk masing-masing organisasi Anda, tetapi Anda tidak dapat mengubah kunci KMS setelah Anda memilihnya.

> **Important**
>
> Amazon hanya WorkMail mendukung kunci KMS enkripsi simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi data di Amazon. WorkMail Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Untuk menemukan kunci KMS untuk organisasi Anda, gunakan entri AWS CloudTrail log yang merekam panggilan keAWS KMS.

Kunci enkripsi unik untuk setiap kotak pesan

Saat Anda membuat kotak pesan baru, Amazon WorkMail menghasilkan kunci enkripsi simetris Advanced Encryption Standard (AES) 256-bit yang unik untuk kotak pesan, yang dikenal sebagai kunci kotak pesan, di luar kotak pesan. AWS KMS Amazon WorkMail menggunakan kunci kotak pesan untuk melindungi kunci enkripsi untuk setiap pesan di kotak pesan.

Untuk melindungi kunci kotak pesan, Amazon WorkMail memanggil AWS KMS untuk mengenkripsi kunci kotak pesan di bawah kunci KMS untuk organisasi. Kemudian ini menyimpan kunci kotak pesan yang dienkripsi dalam metadata kotak pesan.

> **Note**
>
> Amazon WorkMail menggunakan kunci enkripsi kotak pesan simetris untuk melindungi kunci pesan. Sebelumnya, Amazon WorkMail melindungi setiap kotak surat dengan key pair asimetris. Ini menggunakan kunci publik untuk mengenkripsi setiap kunci pesan dan kunci pribadi untuk mendekripsinya. Kunci kotak surat pribadi dilindungi oleh kunci KMS untuk organisasi. Kotak pesan yang ada mungkin masih menggunakan pasangan kunci kotak pesan asimetris. Perubahan ini tidak memengaruhi keamanan kotak surat atau pesannya.

Kunci enkripsi unik untuk setiap pesan

Saat pesan ditambahkan ke kotak pesan, Amazon WorkMail menghasilkan kunci enkripsi simetris AES 256-bit yang unik untuk pesan di luar pesan. AWS KMS Ini menggunakan kunci pesan ini untuk mengenkripsi pesan. Amazon WorkMail mengenkripsi kunci pesan di bawah kunci kotak surat dan menyimpan kunci pesan terenkripsi dengan pesan. Kemudian, ia mengenkripsi kunci kotak surat di bawah kunci KMS untuk organisasi.

Membuat kotak pesan baru

Saat Amazon WorkMail membuat kotak pesan baru, Amazon menggunakan proses berikut untuk menyiapkan kotak pesan untuk menyimpan pesan terenkripsi.

- Amazon WorkMail menghasilkan kunci enkripsi simetris AES 256-bit yang unik untuk kotak surat di luar. AWS KMS
- Amazon WorkMail menyebut operasi AWS KMS Enkripsi. Ini melewati kunci kotak surat dan pengidentifikasi AWS KMS key untuk organisasi. AWS KMSmengembalikan ciphertext dari kunci kotak surat yang dienkripsi di bawah kunci KMS.
- Amazon WorkMail menyimpan kunci kotak surat terenkripsi dengan metadata kotak surat.

Mengenkripsi pesan kotak pesan

Untuk mengenkripsi pesan, Amazon WorkMail menggunakan proses berikut.

1. Amazon WorkMail menghasilkan kunci simetris AES 256-bit yang unik untuk pesan tersebut. Ini menggunakan kunci data plaintext dan algoritme Advanced Encryption Standard (AES) untuk mengenkripsi pesan di luar AWS KMS.
2. Untuk melindungi kunci pesan di bawah kunci kotak surat, Amazon WorkMail perlu mendekripsi kunci kotak surat, yang selalu disimpan dalam bentuk terenkripsi.

   Amazon WorkMail memanggil operasi AWS KMS Dekripsi dan meneruskan kunci kotak surat terenkripsi. AWS KMSmenggunakan kunci KMS untuk organisasi untuk mendekripsi kunci kotak pesan dan mengembalikan kunci kotak pesan teks biasa ke Amazon. WorkMail
3. Amazon WorkMail menggunakan kunci kotak pesan teks biasa dan algoritma Advanced Encryption Standard (AES) untuk mengenkripsi kunci pesan di luar. AWS KMS
4. Amazon WorkMail menyimpan kunci pesan terenkripsi dalam metadata pesan terenkripsi sehingga tersedia untuk mendekripsi.

Mendekripsi pesan kotak pesan

Untuk mendekripsi pesan, Amazon WorkMail menggunakan proses berikut.

1. Amazon WorkMail memanggil operasi AWS KMS Dekripsi dan meneruskan kunci kotak surat terenkripsi. AWS KMSmenggunakan kunci KMS untuk organisasi untuk mendekripsi kunci kotak pesan dan mengembalikan kunci kotak pesan teks biasa ke Amazon. WorkMail

2. Amazon WorkMail menggunakan kunci kotak pesan teks biasa dan algoritma Advanced Encryption Standard (AES) untuk mendekripsi kunci pesan terenkripsi di luar. AWS KMS
3. Amazon WorkMail menggunakan kunci pesan teks biasa untuk mendekripsi pesan terenkripsi.

Melakukan cache tombol kotak pesan

Untuk meningkatkan kinerja dan meminimalkan panggilan keAWS KMS, Amazon WorkMail menyimpan setiap kunci kotak pesan teks biasa untuk setiap klien secara lokal hingga satu menit. Pada akhir periode caching, kunci kotak pesan akan dihapus. Jika kunci kotak surat untuk klien tersebut diperlukan selama periode caching, Amazon WorkMail bisa mendapatkannya dari cache alih-alih menelepon. AWS KMS Kunci kotak pesan dilindungi dalam cache dan tidak pernah ditulis ke disk dalam plaintext.

Mengotorisasi penggunaan kunci KMS

Saat Amazon WorkMail menggunakan operasi kriptografi AWS KMS key dalam, Amazon bertindak atas nama administrator kotak surat.

Untuk menggunakan rahasia atas nama Anda, administrator harus memiliki izin berikut. AWS KMS key Anda dapat menentukan izin yang diperlukan ini dalam kebijakan IAM atau kebijakan kunci.

- `kms:Encrypt`
- `kms:Decrypt`
- `kms:CreateGrant`

Untuk mengizinkan kunci KMS digunakan hanya untuk permintaan yang berasal dari Amazon WorkMail, Anda dapat menggunakan kunci ViaService kondisi kms: dengan nilainya. `workmail.<region>.amazonaws.com`

Anda juga dapat menggunakan kunci atau nilai dalam konteks enkripsi sebagai syarat untuk menggunakan kunci KMS untuk operasi kriptografi. Misalnya, Anda dapat menggunakan operator ketentuan string di IAM atau dokumen kebijakan kunci atau menggunakan batasan hibah dalam hibah.

Kebijakan utama untuk Kunci yang dikelola AWS

Kebijakan utama Kunci yang dikelola AWS untuk Amazon WorkMail memberi pengguna izin untuk menggunakan kunci KMS untuk operasi tertentu hanya ketika Amazon WorkMail membuat

permintaan atas nama pengguna. Kebijakan kunci tidak mengizinkan pengguna untuk menggunakan kunci KMS secara langsung.

Kebijakan utama ini, seperti kebijakan semua Kunci yang dikelola AWS, ditetapkan oleh layanan. Anda tidak dapat mengubah kebijakan kunci, tetapi Anda dapat melihatnya kapan saja. Untuk rincian selengkapnya, lihat Melihat kebijakan kunci.

Pernyataan kebijakan dalam kebijakan kunci memiliki efek sebagai berikut:

- Izinkan pengguna di akun dan Wilayah untuk menggunakan kunci KMS untuk operasi kriptografi dan untuk membuat hibah, tetapi hanya ketika permintaan datang dari Amazon WorkMail atas nama mereka. Kunci kondisi `kms:ViaService` memberlakukan pembatasan ini.
- Memungkinkan Akun AWS untuk membuat kebijakan IAM yang memungkinkan pengguna untuk melihat properti kunci KMS dan mencabut hibah.

Berikut ini adalah kebijakan utama untuk contoh Kunci yang dikelola AWS untuk 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" : "*"
  } ]
}
```

Menggunakan hibah untuk mengotorisasi Amazon WorkMail

Selain kebijakan utama, Amazon WorkMail menggunakan hibah untuk menambahkan izin ke kunci KMS untuk setiap organisasi. Untuk melihat hibah pada kunci KMS di akun Anda, gunakan operasi. ListGrants

Amazon WorkMail menggunakan hibah untuk menambahkan izin berikut ke kunci KMS untuk organisasi.

- Tambahkan `kms:Encrypt` izin untuk mengizinkan Amazon WorkMail mengenkripsi kunci kotak surat.
- Tambahkan `kms:Decrypt` izin untuk mengizinkan Amazon menggunakan kunci KMS WorkMail untuk mendekripsi kunci kotak surat. Amazon WorkMail memerlukan izin ini dalam hibah karena permintaan untuk membaca pesan kotak pesan menggunakan konteks keamanan pengguna yang membaca pesan. Permintaan tidak menggunakan kredensial dari Akun AWS. Amazon WorkMail membuat hibah ini saat Anda memilih kunci KMS untuk organisasi.

Untuk membuat hibah, Amazon WorkMail memanggil CreateGrantatas nama pengguna yang membuat organisasi. Izin untuk membuat hibah berasal dari kebijakan kunci. Kebijakan ini memungkinkan pengguna akun untuk memanggil `CreateGrant` kunci KMS untuk organisasi saat Amazon WorkMail membuat permintaan atas nama pengguna yang berwenang.

Kebijakan kunci juga memungkinkan root akun untuk mencabut hibah pada. Kunci yang dikelola AWS Namun, jika Anda mencabut hibah, Amazon WorkMail tidak dapat mendekripsi data terenkripsi di kotak pesan Anda.

Konteks WorkMail enkripsi Amazon

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS secara kriptografi mengikat konteks enkripsi untuk data terenkripsi tersebut. Untuk mendekripsi data, Anda harus meneruskan konteks enkripsi yang sama.

Amazon WorkMail menggunakan format konteks enkripsi yang sama di semua operasi AWS KMS kriptografi. Anda dapat menggunakan konteks enkripsi untuk mengidentifikasi operasi kriptografi ini dalam catatan audit dan log, seperti AWS CloudTrail, dan sebagai syarat untuk otorisasi dalam kebijakan dan bantuan.

Dalam permintaan Enkripsi dan Dekripsi ke, AWS KMS Amazon WorkMail menggunakan konteks enkripsi di mana kuncinya berada `aws:workmail:arn` dan nilainya adalah Nama Sumber Daya Amazon (ARN) organisasi.

```
"aws:workmail:arn":"arn:aws:workmail:region:account ID:organization/organization ID"
```

Sebagai contoh, konteks enkripsi berikut mencakup contoh organisasi ARN di Wilayah AS Timur (Ohio) (`us-east-2`).

```
"aws:workmail:arn":"arn:aws:workmail:us-east-2:111122223333:organization/
m-68755160c4cb4e29a2b2f8fb58f359d7"
```

Memantau WorkMail interaksi Amazon dengan AWS KMS

Anda dapat menggunakan AWS CloudTrail dan Amazon CloudWatch Logs untuk melacak permintaan yang WorkMail dikirimkan AWS KMS Amazon atas nama Anda.

Enkripsi

Saat Anda membuat kotak pesan baru, Amazon akan membuat WorkMail kunci kotak pesan dan panggilan AWS KMS untuk mengenkripsi kunci kotak pesan. Amazon WorkMail mengirimkan permintaan Enkripsi AWS KMS dengan kunci kotak pesan teks biasa dan pengenal untuk kunci KMS organisasi Amazon. WorkMail

Peristiwa yang mencatat operasi `Encrypt` serupa dengan peristiwa contoh berikut. Pengguna adalah WorkMail layanan Amazon. Parameter termasuk ID kunci KMS (`keyId`) dan konteks enkripsi untuk WorkMail organisasi Amazon. Amazon WorkMail juga melewati kunci kotak surat, tetapi itu tidak dicatat dalam CloudTrail log.

```
{
    "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"
}
```


Dekripsi

Saat menambahkan, melihat, atau menghapus pesan kotak pesan, Amazon WorkMail meminta AWS KMS untuk mendekripsi kunci kotak pesan. Amazon WorkMail mengirimkan permintaan Dekripsi AWS KMS dengan kunci kotak surat terenkripsi dan pengenal untuk kunci KMS organisasi Amazon. WorkMail

Peristiwa yang mencatat operasi `Decrypt` serupa dengan peristiwa contoh berikut. Pengguna adalah WorkMail layanan Amazon. Parameter termasuk kunci kotak surat terenkripsi (sebagai gumpalan ciphertext), yang tidak direkam dalam log, dan konteks enkripsi untuk organisasi Amazon. WorkMail AWS KMSmemperoleh ID kunci KMS dari ciphertext.

```
{
    "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"
}
```


Bagaimana WorkSpaces menggunakan AWS KMS

Anda dapat menggunakan WorkSpacesuntuk menyediakan desktop berbasis cloud (a WorkSpace) untuk setiap pengguna akhir Anda. Ketika Anda meluncurkan yang baru WorkSpace, Anda dapat memilih untuk mengenkripsi volumenya dan memutuskan mana yang AWS KMS keyakan digunakan

untuk enkripsi. Anda dapat memilih Kunci yang dikelola AWSfor WorkSpaces (aws/workspaces) atau kunci yang dikelola pelanggan simetris.

> **Important**
>
> WorkSpaces hanya mendukung kunci KMS enkripsi simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi volume dalam file. WorkSpaces Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat. Mengidentifikasi kunci KMS asimetris

Untuk informasi selengkapnya tentang membuat WorkSpaces dengan volume terenkripsi, buka Enkripsi WorkSpace di Panduan Administrasi Amazon WorkSpaces .

Topik

- Ikhtisar WorkSpaces enkripsi menggunakan AWS KMS
- WorkSpaces konteks enkripsi
- Memberikan WorkSpaces izin untuk menggunakan kunci KMS atas nama Anda

Ikhtisar WorkSpaces enkripsi menggunakan AWS KMS

Saat Anda membuat WorkSpaces dengan volume terenkripsi, gunakan WorkSpaces Amazon Elastic Block Store (Amazon EBS) untuk membuat dan mengelola volume tersebut. Kedua layanan menggunakan Anda AWS KMS key untuk bekerja dengan volume terenkripsi. Untuk informasi selengkapnya tentang enkripsi volume EBS, lihat dokumentasi berikut:

- Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS dalam panduan ini
- Enkripsi Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Windows

Saat Anda meluncurkan WorkSpaces dengan volume terenkripsi, end-to-end prosesnya bekerja seperti ini:

1. Anda menentukan kunci KMS yang akan digunakan untuk enkripsi serta WorkSpace pengguna dan direktori. Tindakan ini membuat hibah yang memungkinkan WorkSpaces untuk menggunakan kunci KMS Anda hanya untuk ini WorkSpace —yaitu, hanya untuk yang WorkSpace terkait dengan pengguna dan direktori yang ditentukan.

2. WorkSpaces membuat volume EBS terenkripsi untuk WorkSpace dan menentukan kunci KMS untuk digunakan serta pengguna dan direktori volume (informasi yang sama yang Anda tentukan di). Step 1 Tindakan ini membuat hibah yang memungkinkan Amazon EBS menggunakan kunci KMS Anda hanya untuk ini WorkSpace dan volume—yaitu, hanya untuk yang WorkSpace terkait dengan pengguna dan direktori yang ditentukan, dan hanya untuk volume yang ditentukan.
3. Amazon EBS meminta kunci data volume yang dienkripsi di bawah kunci KMS Anda dan menentukan ID WorkSpace pengguna `Sid` dan direktori serta ID volume sebagai konteks enkripsi.
4. AWS KMSmembuat kunci data baru, mengenkripsinya di bawah kunci KMS Anda, dan kemudian mengirimkan kunci data terenkripsi ke Amazon EBS.
5. WorkSpaces menggunakan Amazon EBS untuk melampirkan volume terenkripsi ke Anda. WorkSpace Amazon EBS mengirimkan kunci data terenkripsi AWS KMS dengan `Decrypt`permintaan dan menentukan WorkSpace pengguna, ID direktorinya`Sid`, dan ID volume, yang digunakan sebagai konteks enkripsi.
6. AWS KMSmenggunakan kunci KMS Anda untuk mendekripsi kunci data, dan kemudian mengirimkan kunci data plaintext ke Amazon EBS.
7. Amazon EBS menggunakan kunci data plaintext untuk mengenkripsi semua data pergi ke dan dari volume terenkripsi. Amazon EBS menyimpan kunci data plaintext di memori selama volume dilampirkan ke file. WorkSpace
8. Amazon EBS menyimpan kunci data terenkripsi (diterima diStep 4) dengan metadata volume untuk penggunaan di masa mendatang jika Anda me-reboot atau membangun kembali. WorkSpace
9. Saat Anda menggunakan file AWS Management Console untuk menghapus WorkSpace (atau menggunakan `TerminateWorkspaces`tindakan di WorkSpaces API), WorkSpaces dan Amazon EBS menghentikan hibah yang memungkinkan mereka menggunakan kunci KMS Anda untuk itu. WorkSpace

WorkSpaces konteks enkripsi

WorkSpaces tidak menggunakan Anda AWS KMS key secara langsung untuk operasi kriptografi (seperti `Encrypt`, `Decrypt`, `GenerateDataKey`, dll.), Yang WorkSpaces berarti tidak mengirim permintaan ke AWS KMS yang menyertakan konteks enkripsi. Namun, ketika Amazon EBS meminta kunci data terenkripsi untuk volume terenkripsi WorkSpaces (Step 3dalamIkhtisar WorkSpaces enkripsi menggunakan AWS KMS) Anda dan ketika meminta salinan teks biasa dari kunci data tersebut (Step 5), itu menyertakan konteks enkripsi dalam permintaan. Konteks enkripsi menyediakan

data terautentikasi tambahan(AAD) yang AWS KMS gunakan untuk memastikan integritas data. Konteks enkripsi juga ditulis ke file AWS CloudTrail log Anda, yang dapat membantu Anda memahami mengapa yang AWS KMS key diberikan digunakan. Amazon EBS menggunakan hal berikut ini untuk konteks enkripsi:

- AWS Directory ServicePengguna yang terkait dengan `sid` WorkSpace
- ID direktori AWS Directory Service direktori yang terkait dengan WorkSpace
- ID volume dari volume terenkripsi

Contoh berikut menunjukkan representasi JSON dari konteks enkripsi yang digunakan Amazon EBS:

```
{
  "aws:workspaces:sid-directoryid":
 "[S-1-5-21-277731876-1789304096-451871588-1107]@[d-1234abcd01]",
  "aws:ebs:id": "vol-1234abcd"
}
```

Memberikan WorkSpaces izin untuk menggunakan kunci KMS atas nama Anda

Anda dapat melindungi data ruang kerja Anda di bawah Kunci yang dikelola AWS for WorkSpaces (aws/workspaces) atau kunci yang dikelola pelanggan Jika Anda menggunakan kunci yang dikelola pelanggan, Anda harus memberikan WorkSpaces izin untuk menggunakan kunci KMS atas nama administrator di akun Anda. WorkSpaces The Kunci yang dikelola AWS for WorkSpaces memiliki izin yang diperlukan secara default.

Untuk menyiapkan kunci terkelola pelanggan Anda untuk digunakan WorkSpaces, gunakan prosedur berikut.

1. Tambahkan WorkSpaces administrator ke daftar pengguna kunci dalam kebijakan kunci kunci KMS
2. Berikan izin tambahan WorkSpaces kepada administrator dengan kebijakan IAM

WorkSpaces administrator juga memerlukan izin untuk menggunakannya WorkSpaces. Untuk informasi selengkapnya tentang izin ini, buka Mengontrol Akses ke WorkSpaces Sumber Daya di Panduan WorkSpaces Administrasi Amazon.

Bagian 1: Menambahkan WorkSpaces administrator ke pengguna kunci kunci KMS

Untuk memberi WorkSpaces administrator izin yang mereka butuhkan, Anda dapat menggunakan AWS Management Console atau API. AWS KMS

Untuk menambahkan WorkSpaces administrator sebagai pengguna kunci untuk kunci KMS (konsol)

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
4. Pilih ID kunci atau alias kunci terkelola pelanggan pilihan Anda
5. Pilih tab Kebijakan kunci. Di bawah Pengguna kunci, pilih Tambahkan.
6. Dalam daftar pengguna dan peran IAM, pilih pengguna dan peran yang sesuai dengan WorkSpaces administrator Anda, lalu pilih Lampirkan.

Untuk menambahkan WorkSpaces administrator sebagai pengguna kunci untuk kunci KMS (API) AWS KMS

1. Gunakan GetKeyPolicyoperasi untuk mendapatkan kebijakan kunci yang ada, lalu simpan dokumen kebijakan ke file.
2. Buka dokumen kebijakan di editor teks pilihan Anda. Tambahkan pengguna IAM dan peran yang sesuai dengan WorkSpaces administrator Anda ke pernyataan kebijakan yang memberikan izin kepada pengguna utama. Kemudian simpan filenya.
3. Gunakan PutKeyPolicyoperasi untuk menerapkan kebijakan kunci ke kunci KMS.

Bagian 2: Memberikan WorkSpaces izin tambahan kepada administrator

Jika Anda menggunakan kunci yang dikelola pelanggan untuk melindungi WorkSpaces data Anda, selain izin di bagian pengguna utama dari kebijakan kunci default, WorkSpaces administrator memerlukan izin untuk membuat hibah pada kunci KMS. Juga, jika mereka menggunakan AWS Management Consoleuntuk membuat WorkSpaces dengan volume terenkripsi, WorkSpaces administrator memerlukan izin untuk membuat daftar alias dan kunci daftar. Untuk informasi tentang membuat dan mengedit kebijakan pengguna IAM, lihat Kebijakan Terkelola dan Kebijakan Selaras dalam Panduan Pengguna IAM.

Untuk memberikan izin ini kepada WorkSpaces administrator Anda, gunakan kebijakan IAM. Tambahkan pernyataan kebijakan yang mirip dengan contoh berikut ke kebijakan IAM untuk setiap WorkSpaces administrator. Ganti contoh kunci KMS ARN *arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab* () dengan yang valid. Jika WorkSpaces administrator hanya menggunakan WorkSpaces API (bukan konsol), Anda dapat menghilangkan pernyataan kebijakan kedua dengan izin `"kms:ListAliases"` dan`"kms:ListKeys"`.

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

Memprogram API AWS KMS

Anda dapat menggunakan AWS KMS API untuk membuat dan mengelola kunci KMS dan fitur khusus, seperti toko kunci khusus, dan menggunakan kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Referensi AWS Key Management Service API.

Kode sampel dalam topik berikut menunjukkan cara menggunakan SDK AWS untuk memanggil API AWS KMS.

Untuk informasi tentang menggunakan konsol AWS KMS untuk melakukan beberapa tugas ini, lihat Mengelola kunci.

Topik

- Membuat klien
- Bekerja dengan kunci
- Bekerja dengan alias
- Mengenkripsi dan mendekripsi kunci data
- Bekerja dengan kebijakan kunci
- Bekerja dengan izin
- Menguji panggilan AWS KMS API Anda
- AWS KMSkonsistensi akhirnya

Membuat klien

Untuk menggunakan AWS SDK for Java,, the AWS SDK for .NET, the AWS SDK for Python (Boto3), the AWS SDK for RubyAWS SDK for PHP, atau AWSSDK untuk JavaScript di Node.js untuk menulis kode yang menggunakan AWS Key Management Service(AWS KMS) API, mulailah dengan membuat AWS KMS klien.

Objek klien yang Anda buat digunakan pada kode contoh dalam topik yang mengikuti.

Java

Untuk membuat klien AWS KMS di Java, gunakan pembangun klien.

```
AWSKMS kmsClient = AWSKMSClientBuilder.standard().build();
```

Untuk informasi lebih lanjut tentang pembangun klien Java, lihat sumber daya berikut.

- Pembangun Klien Fluent di Blog Developer AWS
- Membuat Klien Layanan di Panduan Developer AWS SDK for Java
- AWSKMSClientBuilder di Referensi API AWS SDK for Java

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

Untuk membuat klien AWS KMS di PHP, gunakan objek klien AWS KMS, dan tentukan versi `2014-11-01`. Untuk informasi lebih lanjut lihat kelas KMSClient di Referensi API AWS SDK for PHP.

```
// 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();
```

Bekerja dengan kunci

Contoh dalam topik ini menggunakan AWS KMS API untuk membuat, melihat, mengaktifkan, dan menonaktifkan AWS KMS AWS KMS keys, dan untuk menghasilkan kunci data.

Topik

- Membuat kunci KMS
- Menghasilkan kunci data
- Melihat sebuah AWS KMS key
- Mendapatkan ID kunci dan ARN kunci dari kunci KMS
- Mengaktifkan AWS KMS keys
- Menonaktifkan AWS KMS key

Membuat kunci KMS

Untuk membuat AWS KMS key(kunci KMS), gunakan CreateKeyoperasi. Contoh di bagian ini membuat kunci KMS enkripsi simetris. Parameter `Description` yang digunakan dalam contoh ini adalah opsional.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Untuk bantuan dalam membuat kunci KMS di AWS KMS konsol, lihatMembuat kunci.

Java

Untuk detailnya, lihat metode createKey di Referensi API AWS SDK for Java.

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest().withDescription(desc);
CreateKeyResult result = kmsClient.createKey(req);
```

C#

Untuk detailnya, lihat CreateKey metode di 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

Untuk detailnya, lihat `create_key`metode diAWS SDK for Python (Boto3).

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kms_client.create_key(
    Description=desc
)
```

Ruby

Untuk detailnya, lihat metode `create_key`instance di AWS SDK for Ruby.

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kmsClient.create_key({
  description: desc
})
```

PHP

Untuk detailnya, lihat CreateKeymetode di AWS SDK for PHP.

```
// Create a KMS key
//
$desc = "Key for protecting critical data";
```

```
$result = $KmsClient->createKey([
    'Description' => $desc
]);
```

Node.js

Untuk detailnya, lihat properti createKey di SDK AWS JavaScript untuk di Node.js.

```
// Create a KMS key
//
const Description = 'Key for protecting critical data';

kmsClient.createKey({ Description }, (err, data) => {
  ...
});
```

PowerShell

Untuk membuat kunci KMS PowerShell, gunakan New- KmsKey cmdlet.

```
# Create a KMS key

$desc = 'Key for protecting critical data'
New-KmsKey -Description $desc
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Menghasilkan kunci data

Untuk menghasilkan kunci data simetris, gunakan GenerateDataKeyoperasi. Operasi ini mengembalikan kunci data plaintext dan salinan kunci data yang dienkripsi di bawah kunci KMS enkripsi simetris yang Anda tentukan. Anda harus menentukan salah satu `KeySpec` atau `NumberOfBytes` (tetapi tidak keduanya) di setiap perintah.

Untuk bantuan menggunakan kunci data untuk mengenkripsi data, lihat. AWS Encryption SDK Anda juga dapat menggunakan kunci data dalam operasi HMAC.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat generateDataKey metode di Referensi AWS SDK for Java API.

```
// 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#

Untuk detailnya, lihat GenerateDataKey metode di 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

Untuk detailnya, lihat generate_data_keymetode diAWS 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

Untuk detailnya, lihat metode generate_data_keyinstance di 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

Untuk detailnya, lihat GenerateDataKeymetode di 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

Untuk detailnya, lihat generateDataKey properti di AWSSDK untuk JavaScript di 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

Untuk menghasilkan kunci data simetris, gunakan cmdlet New-KMS DataKey.

Dalam output, kunci plaintext (di `Plaintext` properti) dan kunci terenkripsi (di `CiphertextBlob` properti) adalah objek. MemoryStream Untuk mengonversinya menjadi string, gunakan metode `MemoryStream` kelas, atau cmdlet atau fungsi yang mengubah `MemoryStream` objek menjadi string, seperti fungsi ConvertFrom- MemoryStream dan ConvertFrom-Base64 dalam modul Convert.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Melihat sebuah AWS KMS key

Untuk mendapatkan informasi terperinci tentangAWS KMS key, termasuk ARN kunci KMS dan status kunci, gunakan DescribeKeyoperasi.

`DescribeKey` tidak mendapatkan alias. Untuk mendapatkan alias, gunakan ListAliasesoperasi. Sebagai contoh, lihat Bekerja dengan alias.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Untuk bantuan dengan melihat tombol KMS di AWS KMS konsol, lihatMelihat kunci.

Java

Untuk detailnya, lihat metode describeKey di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat DescribeKey metode di 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

Untuk detailnya, lihat `describe_key`metode diAWS 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

Untuk detailnya, lihat metode `describe_key`instance di 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

Untuk detailnya, lihat DescribeKeymetode di 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

Untuk detailnya, lihat properti DescribeKey di SDK AWS JavaScript untuk di 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

Untuk mendapatkan informasi rinci tentang kunci KMS, gunakan Get- KmsKey cmdlet.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mendapatkan ID kunci dan ARN kunci dari kunci KMS

Untuk mendapatkan ID kunci dan ARN kunci dariAWS KMS keys, gunakan ListKeysoperasi. Contoh-contoh ini menggunakan `Limit` parameter opsional, yang menetapkan jumlah maksimum kunci KMS yang dikembalikan dalam setiap panggilan. Untuk bantuan mengidentifikasi kunci KMS dalam AWS KMS operasi, lihatPengidentifikasi kunci () KeyId.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Untuk bantuan dalam menemukan ID kunci dan ARN kunci di konsol AWS KMS, lihat Menemukan ID kunci dan kunci ARN.

Java

Untuk detailnya, lihat metode listKeys di Referensi API AWS SDK for Java.

```
// List KMS keys in this account
//
Integer limit = 10;

ListKeysRequest req = new ListKeysRequest().withLimit(limit);
ListKeysResult result = kmsClient.listKeys(req);
```

C#

Untuk detailnya, lihat ListKeys metode di 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

Untuk detailnya, lihat `list_keys`metode diAWS SDK for Python (Boto3).

```
# List KMS keys in this account

response = kms_client.list_keys(
    Limit=10
)
```

Ruby

Untuk detailnya, lihat metode `list_keys`instance di AWS SDK for Ruby.

```
# List KMS keys in this account

response = kmsClient.list_keys({
  limit: 10
})
```

PHP

Untuk detailnya, lihat ListKeys metode di AWS SDK for PHP.

```
// List KMS keys in this account
//
$limit = 10;

$result = $KmsClient->listKeys([
    'Limit' => $limit,
]);
```

Node.js

Untuk detailnya, lihat properti ListKeys di AWSSDK untuk JavaScript di Node.js.

```
// List KMS keys in this account
//
const Limit = 10;
```

```
kmsClient.listKeys({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Untuk mendapatkan ID kunci dan kunci ARN dari semua kunci KMS di akun dan Wilayah, gunakan Get - cmdlet. KmsKeyList

Untuk membatasi jumlah objek output, contoh ini menggunakan cmdlet Select-Object, bukan parameter `Limit`, yang tidak lagi digunakan dalam cmdlet daftar. Untuk bantuan dengan output pemberian nomor halaman di AWS Tools for PowerShell, lihat Output Pemberian Nomor Halaman dengan AWS Tools for PowerShell.

```
# List KMS keys in this account

$limit = 10
Get-KmsKeyList | Select-Object -First $limit
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mengaktifkan AWS KMS keys

Untuk mengaktifkan dinonaktifkanAWS KMS key, gunakan EnableKeyoperasi.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Untuk bantuan mengaktifkan dan menonaktifkan kunci KMS di konsol, lihat. AWS KMS Mengaktifkan dan menonaktifkan kunci

Java

Untuk detail tentang implementasi Java, lihat metode enableKey di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat EnableKey metode di 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

Untuk detailnya, lihat `enable_key`metode diAWS 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

Untuk detailnya, lihat metode `enable_key`instance di 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

Untuk detailnya, lihat EnableKeymetode di 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

Untuk detailnya, lihat properti EnableKey di SDK AWS JavaScript untuk di 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

Untuk mengaktifkan kunci KMS, gunakan Enable- KmsKey cmdlet.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Menonaktifkan AWS KMS key

Untuk menonaktifkan kunci KMS, gunakan DisableKeyoperasi. Menonaktifkan kunci KMS mencegahnya digunakan dalam operasi kriptografi.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Untuk bantuan mengaktifkan dan menonaktifkan kunci KMS di konsol, lihat. AWS KMS Mengaktifkan dan menonaktifkan kunci

Java

Untuk detailnya, lihat metode disableKey di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat DisableKey metode di 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

Untuk detailnya, lihat disable_keymetode diAWS 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

Untuk detailnya, lihat metode disable_keyinstance di 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

Untuk detailnya, lihat DisableKeymetode di 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

Untuk detailnya, lihat properti DisableKey di SDK untuk di AWS Node.js. JavaScript

```
// 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

Untuk menonaktifkan kunci KMS, gunakan 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Bekerja dengan alias

Contoh dalam topik ini menggunakan AWS KMS API untuk membuat, melihat, memperbarui, dan menghapus alias. Untuk informasi tentang cara alias, lihat the section called “Menggunakan alias”.

Topik

Membuat alias

Ketika Anda membuat AWS KMS key diAWS Management Console, Anda harus membuat alias untuk itu. Namun, CreateKeyoperasi yang membuat kunci KMS tidak membuat alias.

Untuk membuat alias, gunakan CreateAliasoperasi. Alias harus unik di akun dan Wilayah. Anda tidak dapat membuat alias yang dimulai dengan `aws/`. `aws/`Awalan dicadangkan oleh Amazon Web Services untuk Kunci yang dikelola AWS.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode createAlias di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat CreateAlias metode di 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

Untuk detailnya, lihat `create_alias`metode diAWS 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

Untuk detailnya, lihat metode `create_alias`instance di 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

Untuk detailnya, lihat CreateAlias metode di 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

Untuk detailnya, lihat properti createAlias di SDK untuk di AWS Node.js. JavaScript

```
// 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

Untuk membuat alias, gunakan cmdlet New-KMSAlias. Nama alias peka terhadap huruf besar-kecil.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Membuat daftar alias

Untuk membuat daftar alias di akun dan wilayah, gunakan ListAliasesoperasi.

Secara default, perintah ListAliases mengembalikan semua alias di akun dan Wilayah. Ini termasuk alias yang Anda buat dan kaitkan dengan kunci yang dikelola pelanggan Anda, dan alias yang AWS dibuat dan dikaitkan dengan Anda. Kunci yang dikelola AWS Respons mungkin juga menyertakan alias yang tidak memiliki bidang `TargetKeyId`. Ini adalah alias yang AWS telah ditentukan sebelumnya yang telah dibuat tetapi belum dikaitkan dengan kunci KMS.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detail tentang implementasi Java, lihat metode listAliases di Referensi API AWS SDK for Java.

```
// List the aliases in this Akun AWS
//
Integer limit = 10;

ListAliasesRequest req = new ListAliasesRequest().withLimit(limit);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

Untuk detailnya, lihat ListAliases metode di AWS SDK for .NET.

```
// List the aliases in this Akun AWS
//
int limit = 10;
```

```
ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    Limit = limit
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

Untuk detailnya, lihat `list_aliases`metode diAWS SDK for Python (Boto3).

```
# List the aliases in this Akun AWS

response = kms_client.list_aliases(
    Limit=10
)
```

Ruby

Untuk detailnya, lihat metode `list_aliases`instance di AWS SDK for Ruby.

```
# List the aliases in this Akun AWS

response = kmsClient.list_aliases({
  limit: 10
})
```

PHP

Untuk detailnya, lihat metode List Aliases di AWS SDK for PHP.

```
// List the aliases in this Akun AWS
//
$limit = 10;

$result = $KmsClient->listAliases([
    'Limit' => $limit,
]);
```

Node.js

Untuk detailnya, lihat properti ListAliases di SDK AWS JavaScript untuk di Node.js.

```
// List the aliases in this Akun AWS
//
const Limit = 10;
kmsClient.listAliases({ Limit }, (err, data) => {
  ...
});
```

PowerShell

Untuk membuat daftar alias di akun dan Wilayah, gunakan cmdlet AliasListGet-KMS.

Untuk membatasi jumlah objek output, contoh ini menggunakan cmdlet Select-Object, bukan parameter `Limit`, yang tidak lagi digunakan dalam cmdlet daftar. Untuk bantuan dengan output pemberian nomor halaman di AWS Tools for PowerShell, lihat Output Pemberian Nomor Halaman dengan AWS Tools for PowerShell.

```
# List the aliases in this Akun AWS
$limit = 10

$result = Get-KMSAliasList | Select-Object -First $limit
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Untuk daftar hanya alias yang terkait dengan kunci KMS tertentu, gunakan parameter. `KeyId` Nilainya dapat berupa ID kunci atau kunci ARN dari setiap kunci KMS di wilayah tersebut. Anda tidak dapat menentukan nama alias atau ARN alias.

Java

Untuk detail tentang implementasi Java, lihat metode listAliases di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat ListAliases metode di 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

Untuk detailnya, lihat `list_aliases`metode diAWS 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

Untuk detailnya, lihat metode `list_aliases`instance di 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

Untuk detailnya, lihat metode List Aliases di 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

Untuk detailnya, lihat properti ListAliases di SDK AWS JavaScript untuk di 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

Untuk membuat daftar alias untuk kunci KMS, gunakan `KeyId` parameter cmdlet AliasListGet-KMS.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Memperbarui alias

Untuk mengaitkan alias yang ada dengan kunci KMS yang berbeda, gunakan operasi. UpdateAlias

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detail tentang implementasi Java, lihat metode updateAlias di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat UpdateAlias metode di 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

Untuk detailnya, lihat update_aliasmetode diAWS 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

Untuk detailnya, lihat metode update_aliasinstance di 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

Untuk detailnya, lihat UpdateAlias metode di 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

Untuk detailnya, lihat properti updateAlias di SDK untuk di Node.js. AWS JavaScript

```
// 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

Untuk mengubah kunci KMS yang terkait dengan alias, gunakan cmdlet Update-KMSalias. Nama alias peka terhadap huruf besar-kecil.

Cmdlet `Update-KMSAlias` tidak mengembalikan output apa pun. Untuk memverifikasi bahwa perintah berfungsi, gunakan cmdlet Get-KMS AliasList.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Menghapus alias

Untuk menghapus alias, gunakan DeleteAliasoperasi. Menghapus alias tidak berpengaruh pada kunci KMS terkait.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat Metode deleteAlias di Referensi API AWS SDK for Java.

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest req  = new DeleteAliasRequest().withAliasName(aliasName);
kmsClient.deleteAlias(req);
```

C#

Untuk detailnya, lihat DeleteAlias metode di 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

Untuk detailnya, lihat delete_aliasmetode diAWS SDK for Python (Boto3).

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kms_client.delete_alias(
    AliasName=alias_name
)
```

Ruby

Untuk detailnya, lihat metode delete_aliasinstance di AWS SDK for Ruby.

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kmsClient.delete_alias({
  alias_name: alias_name
})
```

PHP

Untuk detailnya, lihat DeleteAlias metode di AWS SDK for PHP.

```
// Delete an alias for a KMS key
//
$aliasName = "alias/projectKey1";

$result = $KmsClient->deleteAlias([
    'AliasName' => $aliasName,
]);
```

Node.js

Untuk detailnya, lihat properti DeleteAlias) di SDK untuk di AWS Node.js. JavaScript

```
// Delete an alias for a KMS key
```

```
//
const AliasName = 'alias/projectKey1';
kmsClient.deleteAlias({ AliasName }, (err, data) => {
  ...
});
```

PowerShell

Untuk menghapus alias, gunakan cmdlet Remove-KMSAlias. Nama alias peka terhadap huruf besar-kecil.

Karena cmdlet ini menghapus alias secara permanen, PowerShell meminta Anda untuk mengonfirmasi perintah. `ConfirmImpact` adalah `High`, sehingga Anda tidak dapat menggunakan `ConfirmPreference` untuk menekan prompt ini. Jika Anda harus menekan prompt konfirmasi, tambahkan parameter umum `Confirm` dengan nilai `$false`, misalnya: `-Confirm:$false`.

Cmdlet `Remove-KMSAlias` tidak mengembalikan output apa pun. Untuk memverifikasi bahwa perintah itu efektif, gunakan cmdlet Get-KMS AliasList.

```
# Delete an alias for a KMS key

$aliasName = 'alias/projectKey1'
Remove-KMSAlias -AliasName $aliasName
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mengenkripsi dan mendekripsi kunci data

Contoh dalam topik ini menggunakan Enkripsi, Dekripsi, dan ReEncryptoperasi di API. AWS KMS

Operasi ini dirancang untuk mengenkripsi dan mendekripsi kunci data. Mereka menggunakan a AWS KMS keysdalam operasi enkripsi dan mereka tidak dapat menerima lebih dari 4 KB (4096 byte) data. Meskipun Anda mungkin menggunakannya untuk mengenkripsi sejumlah kecil data, seperti kata sandi atau kunci RSA, operasi tersebut tidak dirancang untuk mengenkripsi data aplikasi.

Untuk mengenkripsi data aplikasi, gunakan fitur enkripsi sisi server dari layanan AWS, atau pustaka enkripsi sisi klien, seperti AWS Encryption SDK atau klien enkripsi Amazon S3.

Topik

- Mengenkripsi kunci data
- Mendekripsi kunci data
- Mengenkripsi ulang kunci data di bawah yang berbeda AWS KMS key

Mengenkripsi kunci data

Operasi Encrypt dirancang untuk mengenkripsi kunci data, tetapi tidak sering digunakan. GenerateDataKeyWithoutPlaintextOperasi GenerateDataKeydan mengembalikan kunci data terenkripsi. Anda dapat menggunakan metode ini ketika Anda memindahkan data terenkripsi ke Wilayah yang berbeda dan ingin mengenkripsi kunci datanya dengan kunci KMS di Wilayah baru.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode encrypt di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat metode Encrypt di 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

Untuk detailnya, lihat metode enkripsi di 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

Untuk detailnya, lihat metode instans enkripsi di 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

Untuk detailnya, lihat metode Encrypt di 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

Untuk detailnya, lihat properti enkripsi di AWS SDK untuk JavaScript di 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

Untuk mengenkripsi kunci data di bawah kunci KMS, gunakan cmdlet Invoke-KMSENcrypt. Ia mengembalikan ciphertext sebagai `MemoryStream` (System.io. MemoryStream) objek. Anda dapat menggunakan objek `MemoryStream` sebagai input ke cmdlet Invoke-KMSDecrypt.

AWS KMS juga mengembalikan kunci data sebagai objek `MemoryStream`. Dalam contoh ini, untuk mensimulasikan kunci data teks biasa, kami membuat array byte dan menulisnya ke objek `MemoryStream`.

Perhatikan bahwa parameter `Plaintext` dari `Invoke-KMSEncrypt` mengambil array byte (`byte[]`); parameter itu tidak memerlukan objek `MemoryStream`. Dimulai pada AWSPowerShell versi 4.0, parameter di semua AWSPowerShell modul yang mengambil array byte dan `MemoryStream` objek menerima array byte, `MemoryStream` objek, string, array string, dan (System.io. `FileInfo` FileInfo) objek. Anda dapat meneruskan salah satu dari jenis ini ke `Invoke-KMSEncrypt`.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mendekripsi kunci data

Untuk mendekripsi kunci data, gunakan operasi Decrypt.

`ciphertextBlob`Yang Anda tentukan harus berupa nilai `CiphertextBlob` bidang dari GenerateDataKey, GenerateDataKeyWithoutPlaintext, atau Enkripsi respons, atau `PrivateKeyCiphertextBlob` bidang dari GenerateDataKeyPairWithoutPlaintextrespons GenerateDataKeyPairatau. Anda juga dapat menggunakan `Decrypt` operasi untuk mendekripsi data yang dienkripsi di luar AWS KMS oleh kunci publik dalam kunci KMS asimetris.

`KeyId`Parameter tidak diperlukan saat mendekripsi dengan kunci KMS enkripsi simetris. AWS KMSbisa mendapatkan kunci KMS yang digunakan untuk mengenkripsi data dari metadata di gumpalan ciphertext. Tetapi selalu merupakan praktik terbaik untuk menentukan kunci KMS yang Anda gunakan. Praktik ini memastikan bahwa Anda menggunakan kunci KMS yang dimaksud, dan mencegah Anda mendekripsi ciphertext secara tidak sengaja menggunakan kunci KMS yang tidak Anda percayai.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode dekripsi di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat Metode Decrypt di 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

Untuk detailnya, lihat metode dekripsi di 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

Untuk detailnya, lihat metode instans dekripsi di 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

Untuk detailnya, lihat Metode Decrypt di 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

Untuk detailnya, lihat properti dekripsi di AWSSDK untuk JavaScript di 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

Untuk mendekripsi kunci data, gunakan cmdlet Invoke-KMSEncrypt.

Cmdlet ini mengembalikan plaintext sebagai (System.io. `MemoryStream` MemoryStream) objek. Untuk mengonversinya menjadi array byte, gunakan cmdlet atau fungsi yang mengonversi objek `MemoryStream` ke array byte, seperti fungsi dalam modul Konversi.

Karena contoh ini menggunakan ciphertext yang dikembalikan oleh cmdlet enkripsi AWS KMS, ia menggunakan objek `MemoryStream` untuk nilai parameter `CiphertextBlob`. Namun, parameter `CiphertextBlob` dari `Invoke-KMSDecrypt` mengambil array byte (`byte[]`); parameter tidak memerlukan objek `MemoryStream`. Dimulai pada AWSPowerShell versi 4.0, parameter di semua AWSPowerShell modul yang mengambil array byte dan `MemoryStream` objek menerima array byte, `MemoryStream` objek, string, array string, dan (System.io. `FileInfo` FileInfo) objek. Anda dapat meneruskan salah satu dari jenis ini ke `Invoke-KMSDecrypt`.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mengenkripsi ulang kunci data di bawah yang berbeda AWS KMS key

Untuk mendekripsi kunci data terenkripsi, dan kemudian segera mengenkripsi ulang kunci data di bawah yang berbeda, gunakan operasi. AWS KMS key ReEncrypt Operasi dilakukan sepenuhnya di sisi server dalam AWS KMS, jadi mereka tidak pernah mengekspos plaintext Anda di luar AWS KMS.

`ciphertextBlob`Yang Anda tentukan harus berupa nilai `CiphertextBlob` bidang dari GenerateDataKey, GenerateDataKeyWithoutPlaintext, atau Enkripsi respons, atau `PrivateKeyCiphertextBlob` bidang dari GenerateDataKeyPairWithoutPlaintextrespons

GenerateDataKeyPairatau. Anda juga dapat menggunakan `ReEncrypt` operasi untuk mengenkripsi ulang data yang dienkripsi di luar AWS KMS oleh kunci publik dalam kunci KMS asimetris.

`SourceKeyId`Parameter tidak diperlukan saat mengenkripsi ulang dengan kunci KMS enkripsi simetris. AWS KMSbisa mendapatkan kunci KMS yang digunakan untuk mengenkripsi data dari metadata di gumpalan ciphertext. Tetapi selalu merupakan praktik terbaik untuk menentukan kunci KMS yang Anda gunakan. Praktik ini memastikan bahwa Anda menggunakan kunci KMS yang dimaksud, dan mencegah Anda mendekripsi ciphertext secara tidak sengaja menggunakan kunci KMS yang tidak Anda percayai.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode reEncrypt di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat ReEncrypt metode di 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

Untuk detailnya, lihat re_encryptmetode diAWS 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

Untuk detailnya, lihat metode re_encryptinstance di 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

Untuk detailnya, lihat ReEncryptmetode di 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

Untuk detailnya, lihat properti ReEncrypt di SDK AWS JavaScript untuk di 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

Untuk mengenkripsi ulang ciphertext di bawah kunci KMS yang sama atau berbeda, gunakan cmdlet Invoke-KMS. ReEncrypt

Karena contoh ini menggunakan ciphertext yang dikembalikan oleh cmdlet enkripsi AWS KMS, ia menggunakan objek `MemoryStream` untuk nilai parameter `CiphertextBlob`. Namun, parameter `CiphertextBlob` dari `Invoke-KMSReEncrypt` mengambil array byte (`byte[]`); parameter tidak memerlukan objek `MemoryStream`. Dimulai pada AWSPowerShell versi 4.0, parameter di semua AWSPowerShell modul yang mengambil array byte dan `MemoryStream` objek menerima array byte, `MemoryStream` objek, string, array string, dan (System.io. `FileInfo` FileInfo) objek. Anda dapat meneruskan salah satu dari jenis ini ke `Invoke-KMSReEncrypt`.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Bekerja dengan kebijakan kunci

Contoh dalam topik ini menggunakan AWS KMS API untuk melihat dan mengubah kebijakan utamaAWS KMS keys.

Untuk detail tentang cara menggunakan kebijakan utama, kebijakan IAM, dan hibah untuk mengelola akses ke kunci KMS Anda, lihat. Kontrol autentikasi dan akses untuk AWS KMS Untuk mendapatkan bantuan mengenai cara menulis dan memformat dokumen kebijakan JSON, lihat Referensi Kebijakan IAM JSON dalam Panduan Pengguna IAM.

Topik

Mencantumkan nama kebijakan kunci

Untuk mendapatkan nama kebijakan kunci untuk sebuahAWS KMS key, gunakan ListKeyPoliciesoperasi. Satu-satunya nama kebijakan kunci yang diembalikannya adalah default.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detail tentang implementasi Java, lihat listKeyPolicies metode di Referensi AWS SDK for Java API.

```
// 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#

Untuk detailnya, lihat ListKeyPolicies metode di 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

Untuk detailnya, lihat list_key_policiesmetode diAWS 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

Untuk detailnya, lihat metode list_key_policiesinstance di 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

Untuk detailnya, lihat ListKeyPoliciesmetode di 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

Untuk detailnya, lihat listKeyPolicies properti di AWSSDK untuk JavaScript di 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

Untuk mencantumkan nama kebijakan kunci default, gunakan cmdlet Get-KMS KeyPolicyList.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mendapatkan kebijakan kunci

Untuk mendapatkan kebijakan kunci untuk sebuahAWS KMS key, gunakan GetKeyPolicyoperasi.

GetKeyPolicy membutuhkan nama kebijakan. Satu-satunya nama kebijakan yang valid adalah default.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat getKeyPolicy metode di Referensi AWS SDK for Java API.

```
// 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#

Untuk detailnya, lihat GetKeyPolicy metode di 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

Untuk detailnya, lihat get_key_policymetode diAWS 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

Untuk detailnya, lihat metode get_key_policyinstance di 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

Untuk detailnya, lihat GetKeyPolicy metode di 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

Untuk detailnya, lihat getKeyPolicy properti di AWSSDK untuk JavaScript di 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

Untuk mendapatkan kebijakan kunci untuk kunci KMS, gunakan cmdlet KeyPolicyGet-KMS. Cmdlet ini mengembalikan kebijakan kunci sebagai string (System.String) yang dapat Anda gunakan dalam perintah KeyPolicywrite-KMS (). `PutKeyPolicy` Untuk mengonversi kebijakan dalam string JSON menjadi `PSCustomObject` objek, gunakan cmdlet ConvertFrom-JSON.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mengatur kebijakan kunci

Untuk membuat atau mengganti kebijakan kunci untuk kunci KMS, gunakan PutKeyPolicyoperasi.

`PutKeyPolicy` memerlukan nama kebijakan. Satu-satunya nama kebijakan yang valid adalah default.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat putKeyPolicy metode di Referensi AWS SDK for Java API.

```
// 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#

Untuk detailnya, lihat PutKeyPolicy metode di 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

Untuk detailnya, lihat put_key_policymetode diAWS 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

Untuk detailnya, lihat metode put_key_policyinstance di 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

Untuk detailnya, lihat PutKeyPolicy metode di 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

Untuk detailnya, lihat putKeyPolicy properti di AWSSDK untuk JavaScript di 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

Untuk menetapkan kebijakan kunci untuk kunci KMS, gunakan cmdlet KeyPolicyWrite-KMS. Cmdlet ini tidak mengembalikan output apa pun. Untuk memverifikasi bahwa perintah itu efektif, gunakan cmdlet Get-KMS KeyPolicy.

Parameter `Policy` mengambil string. Sertakan string dalam tanda kutip tunggal untuk membuatnya string literal. Anda tidak harus menggunakan karakter lanjutan atau karakter escape dalam string literal.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Bekerja dengan izin

Contoh dalam topik ini menggunakan AWS KMS API untuk membuat, melihat, menghentikan, dan mencabut hibah. AWS KMS keys Untuk detail selengkapnya tentang penggunaan izin di AWS KMS, lihat Hibah di AWS KMS.

Topik

- Membuat izin
- Melihat izin
- Menghentikan izin
- Mencabut izin

Membuat izin

Untuk membuat hibah untukAWS KMS key, gunakan CreateGrantoperasi. Responsnya hanya mencakup ID izin dan token izin. Untuk mendapatkan informasi rinci tentang hibah, gunakan ListGrantsoperasi, seperti yang ditunjukkan padaMelihat izin.

Contoh-contoh ini membuat hibah yang memungkinkan pengguna yang dapat mengambil `ExampleKeyUser` peran untuk memanggil GenerateDataKeyoperasi pada kunci KMS yang diidentifikasi oleh `KeyId` parameter.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode createGrant di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat CreateGrant metode di 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

Untuk detailnya, lihat `create_grant`metode diAWS 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

Untuk detailnya, lihat metode `create_grant`instance di 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

Untuk detailnya, lihat CreateGrantmetode di 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

Untuk detailnya, lihat properti createGrant di SDK AWS JavaScript untuk di 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

Untuk membuat izin, gunakan 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Melihat izin

Untuk mendapatkan informasi terperinci tentang hibah pada kunci KMS, gunakan operasi. ListGrants

> **Note**
>
> Kolom `GranteePrincipal` dalam respons `ListGrants` biasanya berisi perwakilan penerima pemberian izin. Namun, jika perwakilan penerima izin dalam izin adalah layanan AWS, bidang `GranteePrincipal` berisi perwakilan layanan, yang mungkin mewakili beberapa perwakilan penerima izin yang berbeda.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Contoh-contoh ini menggunakan parameter `Limits` opsional, yang menentukan berapa banyak izin yang dikembalikan oleh operasi.

Java

Untuk detail tentang implementasi Java, lihat metode listGrants di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat ListGrants metode di 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

Untuk detailnya, lihat `list_grants`metode diAWS 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

Untuk detailnya, lihat metode `list_grants`instance di 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

Untuk detailnya, lihat ListGrantsmetode di 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

Untuk detailnya, lihat properti ListGrants di SDK AWS JavaScript untuk di 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

Untuk melihat detail semua AWS KMS hibah untuk kunci KMS, gunakan cmdlet GrantListGet-KMS.

Untuk membatasi jumlah objek output, contoh ini menggunakan cmdlet Select-Object, bukan parameter `Limit`, yang tidak lagi digunakan dalam cmdlet daftar. Untuk bantuan dengan output pemberian nomor halaman di AWS Tools for PowerShell, lihat Output Pemberian Nomor Halaman dengan 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Anda harus menentukan kunci KMS di setiap `ListGrants` operasi. Namun, Anda dapat memfilter daftar izin lebih lanjut dengan menentukan ID izin atau perwakilan penerima izin. Contoh berikut hanya mendapatkan hibah untuk kunci KMS di mana `test-engineer` perannya adalah pokok penerima hibah.

Java

Untuk detail tentang implementasi Java, lihat metode listGrants di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat ListGrants metode di 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

Untuk detailnya, lihat `list_grants`metode diAWS 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

Untuk detailnya, lihat metode `list_grants`instance di 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

Untuk detailnya, lihat ListGrantsmetode di 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

Untuk detailnya, lihat properti ListGrants di SDK AWS JavaScript untuk di 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

Untuk melihat detail semua AWS KMS hibah untuk kunci KMS, gunakan cmdlet GrantListGet-KMS.

```
# 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Menghentikan izin

Untuk pensiun hibah untuk kunci KMS, gunakan operasi. RetireGrant Anda harus menghentikan izin untuk menghapus setelah Anda selesai menggunakannya.

Untuk mempensiunkan hibah, berikan token hibah, atau ID hibah dan ID kunci KMS. Untuk operasi ini, ID kunci KMS harus Amazon Resource Name (ARN) dari kunci KMS. Token hibah dikembalikan oleh CreateGrantoperasi. ID hibah dikembalikan oleh CreateGrant dan ListGrantsoperasi.

RetireGrant tidak mengembalikan respons. Untuk memverifikasi bahwa itu efektif, gunakan ListGrantsoperasi.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode pensionGrant di Referensi API AWS SDK for Java.

```
// Retire a grant
//
String grantToken = Place your grant token here;

RetireGrantRequest req = new RetireGrantRequest().withGrantToken(grantToken);
```

```
kmsClient.retireGrant(req);
```

C#

Untuk detailnya, lihat RetireGrant metode di AWS SDK for .NET.

```
// Retire a grant
//
String grantToken = "Place your grant token here";

RetireGrantRequest retireGrantRequest = new RetireGrantRequest()
{
    GrantToken = grantToken
};
kmsClient.RetireGrant(retireGrantRequest);
```

Python

Untuk detailnya, lihat `retire_grant`metode diAWS SDK for Python (Boto3).

```
# Retire a grant

grant_token = Place your grant token here

response = kms_client.retire_grant(
    GrantToken=grant_token
)
```

Ruby

Untuk detailnya, lihat metode `retire_grant`instance di AWS SDK for Ruby.

```
# Retire a grant

grant_token = Place your grant token here

response = kmsClient.retire_grant({
  grant_token: grant_token
})
```

PHP

Untuk detailnya, lihat RetireGrantmetode di AWS SDK for PHP.

```
// Retire a grant
//
$grantToken = 'Place your grant token here';

$result = $KmsClient->retireGrant([
    'GrantToken' => $grantToken,
]);
```

Node.js

Untuk detailnya, lihat properti RetireGrant di SDK AWS JavaScript untuk di Node.js.

```
// Retire a grant
//
const GrantToken = 'Place your grant token here';
kmsClient.retireGrant({ GrantToken }, (err, data) => {
  ...
});
```

PowerShell

Untuk menghentikan izin, gunakan cmdlet Disable-KMSGrant. Untuk mendapatkan token izin, gunakan cmdlet New-KMSGrant. Parameter `GrantToken` mengambil string, jadi Anda tidak perlu mengonversi output yang dikembalikan oleh cmdlet Read-Host.

```
# Retire a grant

$grantToken = Read-Host -Message Place your grant token here
Disable-KMSGrant -GrantToken $grantToken
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Mencabut izin

Untuk mencabut hibah ke kunci KMS, gunakan operasi. RevokeGrant Anda dapat mencabut izin untuk secara eksplisit menolak operasi yang bergantung padanya.

Dalam bahasa yang memerlukan objek klien, contoh-contoh ini menggunakan objek klien AWS KMS yang Anda buat di Membuat klien.

Java

Untuk detailnya, lihat metode revokeGrant di Referensi API AWS SDK for Java.

```
// 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#

Untuk detailnya, lihat RevokeGrant metode di 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);
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools. KeyManagementServicemodul. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Python

Untuk detailnya, lihat revoke_grantmetode diAWS 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

Untuk detailnya, lihat metode revoke_grantinstance di 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

Untuk detailnya, lihat RevokeGrantmetode di 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

Untuk detailnya, lihat properti RevokeGrant di SDK untuk di AWS Node.js. JavaScript

```
// 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

Untuk mencabut izin, gunakan 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
```

Untuk menggunakan AWS KMS PowerShell cmdlet, instal AWS.tools.KeyManagementServicemodul. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Tools for Windows PowerShell.

Menguji panggilan AWS KMS API Anda

Untuk menggunakannyaAWS KMS, Anda harus memiliki kredensi yang AWS dapat digunakan untuk mengautentikasi permintaan API Anda. Kredensialnya harus menyertakan izin untuk mengakses kunci dan alias KMS. Izin ditentukan oleh kebijakan utama, kebijakan IAM, hibah, dan kontrol akses lintas akun. Selain mengontrol akses ke kunci KMS, Anda dapat mengontrol akses ke CloudHSM Anda, dan ke toko kunci khusus Anda.

Anda dapat menentukan parameter `DryRun` API untuk memverifikasi bahwa Anda memiliki izin yang diperlukan untuk menggunakan AWS KMS kunci. Anda juga dapat menggunakan `DryRun` untuk memverifikasi bahwa parameter permintaan dalam panggilan AWS KMS API ditentukan dengan benar.

Topik

- Apa DryRun parameternya?
- Menentukan DryRun dengan API

Apa DryRun parameternya?

`DryRun`adalah parameter API opsional yang Anda tentukan untuk memverifikasi bahwa panggilan AWS KMS API akan berhasil. Gunakan `DryRun` untuk menguji panggilan API Anda, sebelum benar-benar melakukan panggilan keAWS KMS. Anda dapat memverifikasi yang berikut ini.

- Bahwa Anda memiliki izin yang diperlukan untuk menggunakan AWS KMS kunci.
- Bahwa Anda telah menentukan parameter dalam panggilan dengan benar.

AWS KMSmendukung penggunaan `DryRun` parameter dalam tindakan API tertentu:

- CreateGrant
- Dekripsi
- Enkripsi
- GenerateDataKey

- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Tanda
- Verifikasi
- VerifyMac

Menggunakan `DryRun` parameter akan dikenakan biaya dan akan ditagih sebagai permintaan API standar. Untuk informasi lebih lanjut tentang harga AWS KMS, lihat Harga AWS Key Management Service.

Semua permintaan API yang menggunakan `DryRun` parameter berlaku untuk kuota permintaan API dan dapat menghasilkan pengecualian pembatasan jika Anda melebihi kuota permintaan API. Misalnya, memanggil Dekripsi dengan `DryRun` atau tanpa `DryRun` hitungan terhadap kuota operasi kriptografi yang sama. Lihat Permintaan pelambatan AWS KMS untuk mempelajari selengkapnya.

Setiap panggilan ke operasi AWS KMS API ditangkap sebagai peristiwa dan direkam dalam AWS CloudTrail log. Output dari setiap operasi yang menentukan `DryRun` parameter muncul di CloudTrail log Anda. Untuk informasi selengkapnya, lihat Logging panggilan AWS KMS API dengan AWS CloudTrail.

Menentukan DryRun dengan API

Untuk menggunakan`DryRun`, tentukan `–dry-run` parameter dalam AWS CLI perintah dan panggilan AWS KMS API yang mendukung parameter. Ketika Anda melakukannya, AWS KMS akan memverifikasi apakah panggilan Anda akan berhasil. AWS KMSpanggilan yang digunakan `DryRun` akan selalu gagal dan mengembalikan pesan dengan informasi tentang alasan mengapa panggilan gagal. Pesan dapat mencakup pengecualian berikut:

- `DryRunOperationException`- Permintaan akan berhasil jika `DryRun` tidak ditentukan.
- `ValidationException`- Permintaan gagal menentukan parameter API yang salah.

- `AccessDeniedException`- Anda tidak memiliki izin untuk melakukan tindakan API yang ditentukan pada sumber daya KMS.

Misalnya, perintah berikut menggunakan CreateGrantoperasi dan membuat hibah yang memungkinkan pengguna yang berwenang untuk mengambil `keyUserRole` peran untuk memanggil operasi Dekripsi pada kunci KMS simetris tertentu. `DryRun`Parameter ditentukan.

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

AWS KMSkonsistensi akhirnya

AWS KMSAPI mengikuti model konsistensi akhirnya karena sifat sistem yang terdistribusi. Akibatnya, perubahan pada AWS KMS sumber daya mungkin tidak langsung terlihat oleh perintah berikutnya yang Anda jalankan.

Saat Anda melakukan panggilan AWS KMS API, mungkin ada penundaan singkat sebelum perubahan tersedia secara keseluruhanAWS KMS. Biasanya diperlukan waktu kurang dari beberapa detik agar perubahan menyebar ke seluruh sistem, tetapi dalam beberapa kasus dapat memakan waktu beberapa menit. Anda mungkin mendapatkan kesalahan yang tidak terduga, seperti a `NotFoundException` atau`InvalidStateException`, selama waktu ini. Misalnya, AWS KMS mungkin mengembalikan `NotFoundException` jika Anda menelepon `GetParametersForImport` segera setelah meneleponCreateKey.

Kami menyarankan Anda mengonfigurasi strategi coba lagi pada AWS KMS klien Anda untuk mencoba kembali operasi secara otomatis setelah masa tunggu singkat. Untuk informasi selengkapnya, lihat Mencoba lagi perilaku di AWS SDK dan Panduan Referensi Alat.

Untuk panggilan API terkait hibah, Anda dapat menggunakan token hibah untuk menghindari kemungkinan penundaan dan segera menggunakan izin dalam hibah. Untuk informasi lebih lanjut, lihat Konsistensi akhir (untuk hibah).

Referensi

Referensi berikut memberikan informasi yang berguna tentang penggunaan dan pengelolaan kunci KMS.

- Referensi tipe kunci. Daftar jenis kunci KMS yang mendukung setiap operasi AWS KMS API.

  Untuk menemukan: Dapatkah saya mengaktifkan dan menonaktifkan kunci KMS penandatanganan RSA?

- Tabel status kunci. Menunjukkan bagaimana status kunci kunci KMS memengaruhi penggunaannya dalam operasi AWS KMS API.

  Untuk menemukan: Dapatkah saya mengubah alias kunci KMS yang tertunda penghapusan?

- AWS KMSReferensi izin API. Memberikan informasi tentang izin yang diperlukan untuk setiap operasi AWS KMS API.

  Untuk menemukan: Dapatkah saya menjalankan GetKeyPolicykunci di AWS akun yang berbeda? Dapatkah saya mengizinkan `kms:Decrypt` izin dalam kebijakan IAM?

  - ViaService referensi. Daftar AWS layanan yang mendukung kunci `kms:ViaService` kondisi.

    Untuk menemukan: Dapatkah saya menggunakan kunci `kms:ViaService` kondisi untuk mengizinkan izin hanya jika berasal dari Amazon ElastiCache? Bagaimana dengan Amazon Neptunus?

- AWS KMSharga. Daftar dan menjelaskan harga kunci KMS.

  Untuk menemukan: Berapa biaya untuk menggunakan kunci asimetris saya?

- AWS KMSpermintaan kuota. Daftar kuota per detik untuk permintaan AWS KMS API di setiap akun dan Wilayah.

  Untuk menemukan: Berapa banyak permintaan Dekripsi yang dapat saya jalankan di setiap detik? Berapa banyak permintaan Dekripsi yang dapat saya jalankan pada kunci KMS di toko kunci khusus saya?

- AWS KMSkuota sumber daya. Daftar kuota pada sumber AWS KMS daya.

  Untuk menemukan: Berapa banyak kunci KMS yang dapat saya miliki di setiap Wilayah akun saya? Berapa banyak alias yang dapat saya miliki pada setiap kunci KMS?

- AWSlayanan terintegrasi dengan AWS KMS. Daftar AWS layanan yang menggunakan kunci KMS untuk melindungi sumber daya yang mereka buat, simpan, dan kelola.

  Untuk menemukan: Apakah Amazon Connect menggunakan kunci KMS untuk melindungi sumber daya Connect saya?

Riwayat dokumen

Topik ini menjelaskan pembaruan yang signifikan untuk Panduan Pengembang AWS Key Management Service .

Topik

Pembaruan terkini

Tabel berikut menjelaskan perubahan signifikan pada dokumentasi ini sejak Januari 2018. Selain perubahan besar yang ditampilkan di sini, kami juga sering memperbarui dokumentasi untuk memperbaiki deskripsi dan contoh, serta membahas umpan balik yang Anda kirimkan kepada kami. Untuk diberitahu tentang perubahan signifikan, berlangganan umpan RSS.

Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Perubahan	Deskripsi	Tanggal
Pembaruan untuk rotasi kunci	Menambahkan dukungan untuk periode rotasi khusus untuk rotasi kunci otomatis, rotasi kunci sesuai permintaan, dan visibilitas ke rotasi material utama Anda.	April 12, 2024
Pembaruan kebijakan terkelola	Menambahkan izin baru ke `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` yang memungkinkan AWS KMS untuk memantau perubahan di VPC yang berisi cluster AWS CloudHSM	10 November 2023

	Anda AWS KMS sehingga dapat memberikan pesan kesalahan yang jelas jika terjadi kegagalan.	
Pembaruan fitur	Menambahkan dukungan untuk parameter `DryRun` API.	5 Juli 2023
Pembaruan fitur	Menambahkan dukungan untuk mengimpor bahan kunci untuk semua jenis AWS KMS kunci, kecuali toko kunci khusus.	Juni 5, 2023
Pembaruan fitur	Pembaruan untuk AWS KMS API untuk Nitro Enclave	10 Maret 2023
Pembaruan fitur	Algoritma `RSAES_PKCS1_V1_5` pembungkus tidak digunakan lagi. AWS KMS akan mengakhiri semua dukungan untuk pada 1 Oktober `RSAES_PKCS1_V1_5` 2023 sesuai dengan panduan manajemen kunci kriptografi dari National Institute of Standards and Technology (NIST). Kami menyarankan Anda segera mulai menggunakan algoritma pembungkus yang berbeda.	28 Februari 2023
Pembaruan fitur	Menambahkan dukungan untuk toko kunci Eksternal, fitur yang memungkinkan Anda melindungi AWS sumber daya Anda menggunakan kunci kriptografi di luar. AWS	29 November 2022

Perubahan kuota	Peningkatan kuota AWS KMS keys sumber daya menjadi 100.000 kunci KMS di setiap akun dan Wilayah.	8 Juli 2022
Pembaruan fitur	Ditambahkan dukungan untuk kunci HMAC KMS di lebih Wilayah AWS	8 Juli 2022
Topik baru	Menambahkan Ketahanan dalam AWS Key Management Service topik ke bagian Keamanan Panduan AWS KMS Pengembang.	14 Juni 2022
Fitur baru	Menambahkan dukungan untuk AWS KMS kunci dan operasi API yang menghasilkan dan memverifikasi kode HMAC.	19 April 2022
Perubahan dokumentasi	Ganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS.	Agustus 30, 2021
Fitur baru	Menambahkan dukungan untuk kunci Multi-wilayah, satu set kunci KMS yang dapat dioperasikan di Wilayah berbeda yang memiliki ID kunci dan bahan kunci yang sama. Anda dapat menggunakan kunci multi-Wilayah untuk mengenkripsi data dalam satu Wilayah dan mendekripsi data di Wilayah yang berbeda.	8 Juni 2021

Fitur baru	Penambahan dukungan untuk kontrol akses berbasis atribut atau attribute based access control (ABAC). Anda dapat menggunakan tag dan alias untuk mengontrol akses ke Anda AWS KMS keys.	17 Desember 2020
Fitur baru	Penambahan dukungan untuk kebijakan VPC endpoint.	9 Juli 2020
Konten baru	Menjelaskan sifat keamanan AWS KMS.	18 Juni 2020
Fitur baru	Menambahkan dukungan untuk kunci data asimetris AWS KMS keys dan asimetris.	25 November 2019
Fitur yang diperbarui	Anda dapat melihat kebijakan utama Kunci yang dikelola AWS di AWS KMS konsol. Fitur ini dulu terbatas pada kunci yang dikelola pelanggan.	15 November 2019
Fitur baru	Menjelaskan cara menggunakan algoritme pertukaran kunci pasca-kuantum hibrida di TLS untuk panggilan Anda ke AWS KMS.	04 November 2019
Perubahan kuota	Meningkatkan kuota sumber daya untuk beberapa API yang mengelola kunci KMS.	18 September 2019
Perubahan kuota	Mengubah kuota sumber daya untuk kunci KMS, alias, dan hibah per kunci KMS.	27 Maret 2019

Perubahan kuota	Mengubah kuota permintaan per detik bersama untuk operasi kriptografi yang digunakan AWS KMS keys di penyimpanan kunci kustom.	7 Maret 2019
Fitur baru	Menjelaskan cara membuat dan mengelola toko kunci AWS KMS khusus. Setiap toko kunci didukung oleh AWS CloudHSM cluster yang Anda miliki dan kendalikan.	26 November 2018
Konsol baru	Menjelaskan cara menggunakan AWS KMS konsol baru, yang independen dari konsol IAM. Konsol asli, dan petunjuk untuk menggunakannya, akan tetap tersedia untuk jangka waktu singkat guna memberi Anda waktu untuk membiasakan diri dengan konsol baru.	7 November 2018
Perubahan kuota	Mengubah kuota permintaan bersama untuk penggunaan. AWS KMS keys	21 Agustus 2018
Konten baru	Menjelaskan bagaimana AWS Secrets Manager menggunakan AWS KMS kunci untuk mengenkripsi nilai rahasia dalam rahasia.	13 Juli 2018

Konten baru	Menjelaskan bagaimana DynamoDB AWS KMS AWS KMS keys menggunakan untuk mendukung opsi enkripsi sisi servernya.	23 Mei 2018
Fitur baru	Menjelaskan cara menggunakan endpoint pribadi di VPC Anda untuk terhubung langsung AWS KMS, alih-alih menghubungkan melalui internet.	22 Januari 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting pada Panduan AWS Key Management Service Pengembang sebelum 2018.

Anda mungkin perlu menggulir secara horizontal atau vertikal untuk melihat semua data dalam tabel ini.

Perubahan	Deskripsi	Tanggal
Konten baru	Penambahan dokumentasi tentang Tombol penandaan.	Februari 15, 2017
Konten baru	Penambahan dokumentasi tentang Memantau AWS KMS keys dan Pemantauan CloudWatch dengan Amazon.	Agustus 31, 2016
Konten baru	Penambahan dokumentasi tentang Materi kunci yang diimpor.	Agustus 11, 2016
Konten baru	Ditambahkan dokumentasi berikut:Kebijakan	Juli 5, 2016

Perubahan	Deskripsi	Tanggal
	IAM,Referensi izin, danKunci syarat.	
Pembaruan	Bagian yang diperbarui dari dokumentasi di bab Kontrol autentikasi dan akses.	Juli 5, 2016
Pembaruan	Pembaruan halaman Kuota untuk mencerminkan kuota default baru.	31 Mei 2016
Pembaruan	Pembaruan halaman Kuota untuk mencerminkan kuota default baru, dan pembaruan dokumentasi token bantuan untuk meningkatkan kejelasan dan akurasi.	April 11, 2016
Konten baru	Penambahan dokumenta si tentang Mengizinkan beberapa prinsipal IAM untuk mengakses kunci KMS dan Menggunakan syarat alamat IP.	Februari 17, 2016
Pembaruan	Pembaruan halaman Kebijakan utama di AWS KMS dan Mengubah kebijakan kunci untuk meningkatkan kejelasan dan akurasi.	Februari 17, 2016
Pembaruan	Pembaruan halaman topik Mengelola kunci untuk meningkatkan kejelasan.	Januari 5, 2016

Perubahan	Deskripsi	Tanggal
Konten baru	Penambahan dokumentasi tentang Bagaimana AWS CloudTrail menggunakan AWS KMS.	November 18, 2015
Konten baru	Penambahan instruksi untuk Mengubah kebijakan kunci.	November 18, 2015
Pembaruan	Pembaruan dokumentasi tentang Bagaimana Amazon Relational Database Service (Amazon RDS) menggunakan AWS KMS.	November 18, 2015
Konten baru	Penambahan dokumenta si tentang Bagaimana WorkSpaces menggunakan AWS KMS.	November 6, 2015
Pembaruan	Pembaruan halaman Kebijakan utama di AWS KMS untuk meningkatkan kejelasan .	Oktober 22, 2015
Konten baru	Penambahan dokumenta si tentang Menghapus AWS KMS keys, termasuk dokumentasi pendukung tentang Membuat alarm dan Menentukan penggunaan kunci KMS di masa lalu.	Oktober 15, 2015
Konten baru	Penambahan dokumentasi tentang Menentukan akses ke AWS KMS keys.	Oktober 15, 2015

Perubahan	Deskripsi	Tanggal
Konten baru	Penambahan dokumentasi tentang Status AWS KMS kunci kunci.	Oktober 15, 2015
Konten baru	Penambahan dokumenta si tentang Bagaimana Amazon Simple Email Service (Amazon SES) menggunakan AWS KMS.	1 Oktober 2015
Pembaruan	Pembaruan halaman Kuota untuk menjelaskan kuota permintaan baru.	Agustus 31, 2015
Konten baru	Menambahkan informasi tentang biaya untuk menggunakan AWS KMS. Lihat AWS KMS Harga.	Agustus 14, 2015
Konten baru	Menambahkan kuota permintaan ke. AWS KMS Kuota	Juni 11, 2015
Konten baru	Penambahan contoh kode Java baru yang menunjukk an penggunaan operasi `UpdateAlias` . Lihat Memperbarui alias.	1 Juni 2015
Perbarui	AWS Key Management Service Memindahkan tabel daerah ke Referensi Umum AWS.	29 Mei 2015

Perubahan	Deskripsi	Tanggal
Konten baru	Penambahan dokumentasi tentang Bagaimana Amazon EMR menggunakan AWS KMS.	Januari 28, 2015
Konten baru	Penambahan dokumentasi tentang Bagaimana Amazon WorkMail menggunakan AWS KMS.	Januari 28, 2015
Konten baru	Penambahan dokumentasi tentang Bagaimana Amazon Relational Database Service (Amazon RDS) menggunakan AWS KMS.	Januari 6, 2015
Konten baru	Penambahan dokumenta si tentang Bagaimana Amazon Elastic Transcoder menggunakan AWS KMS.	November 24, 2014
Panduan baru	MemperkenalkanPanduan Pengembang AWS Key Management Service .	November 12, 2014

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.